The Local Analysis and Prediction System (LAPS)

This LAPS README file is viewable on the WWW via the LAPS home page at http://laps.noaa.gov/

Unoffical version

Date: November 2, 2012

Update November 23, 2012

Contents

List of Figures

Chapter 1

General LAPS Info

Below is a description of the tar file containing the LAPS data ingest and analysis code. The predictive component of LAPS (MM5, RAMS/SFM, ETA) is set up separately (see section 4.4). Please note that GSD provides support for LAPS software only if a prior agreement is made to that effect. Additionally, questions concerning LAPS must be asked in reference to the latest released tar file; we cannot support older versions of LAPS code. It is also recommended that LAPS users try to take advantage of the latest LAPS updates by periodically importing a fresh tar file every few months or so. Please check the LAPS Software Page at http://laps.noaa.gov/cgi/LAPS_SOFTWARE.cgi for information about recent releases.
A flow chart for the LAPS software can be viewed in figure 1.1 or can be found at:
http://laps.noaa.gov/doc/Slide1.png.

1.1 LAPS Software Disclaimer

Open Source License/Disclaimer, Forecast Systems Laboratory NOAA/OAR/GSD, 325 Broadway Boulder, CO 80305
This software is distributed under the Open Source Definition, which may be found at
http://www.opensource.org/.
In particular, redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain this notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must provide access to this notice, this list of conditions and the following disclaimer, and the underlying source code.
- All modifications to this software must be clearly documented, and are solely the responsibility of the agent making the modifications.
- If significant modifications or enhancements are made to this software, the GSD Software Policy Manager (softwaremgr.fsl@noaa.gov) should be notified.

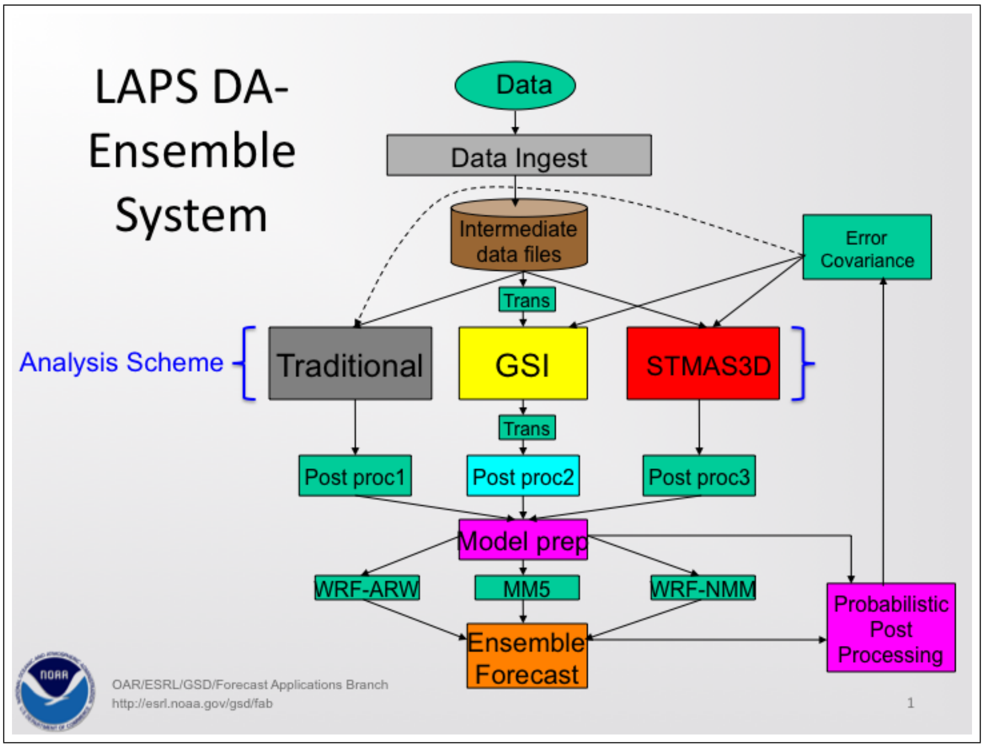

Figure 1.1: Flow chart of the LAPS software

Chapter 2

Installing LAPS Code

2.1 UNIX System Requirements

Supported UNIX platforms include...

```
IBM rs6000       AIX4.3          NFS mounted disks should be mounted
                                 with NFS version 2 instead of 3.
HP               HPUX 10.20      Requires f90
SunOS (Solaris) 5.6              Requires f90
IRIX64           6.5             Requires f90
DEC (Alpha)
LINUX (i386,i686,x86_64)         pgf90 is suggested,
                                 ifort & gfortran being tested
LINUX (MacOS-X)                  pgf90 is being tested
LINUX (Alpha)                    fort is suggested
```

We are working on adding more supported platforms. We welcome suggestions on how to modify LAPS for other platforms/versions.

Note that we cannot guarantee the portability of LAPS to all of these other platforms (e.g. Windows NT).

2.1.1 NetCDF Library

The netCDF package is required for laps, it is available via internet at
http://www.unidata.ucar.edu/packages/netcdf/ netCDF 3.3.1 or a higher 3.x is required. Note that netCDF versions 4.x aren't yet supported. Once netCDF is properly installed, check that the 'ncdump' and 'ncgen' programs are in your path (e.g. 'which ncdump'), so that 'configure' will find them and provide the laps package with the proper path. The path setting will also help LAPS programs to run properly. Note that this path specficiation will override anything supplied with the '–netcdf' command line argument.
NetCDF is a general format structure. The detailed format of each data file is self-describing (via 'ncdump'), and is mirrored in a separate static file called the CDL. This CDL can be GSD's

version or someone elses.
If possible, check that your netCDF library is built to be compatible with the same FORTRAN compiler that you are using.
The netCDF library contains C routines that are to be linked with the LAPS FORTRAN routines. Please see the discussion in paragraph of section 2.2.2.1 for details on troubleshooting this.

2.1.2 Perl

The perl package is also required for laps, it is available via internet at any perl site such as http://www.perl.com. Perl 5.003 or higher is required. Check that 'perl' is in your path (e.g. 'which perl').

2.1.3 Make

Laps Makefiles work best by using gnu make (version 3.75 or higher). This is downloadable from gnu sites such as the following URL: http://www.gnu.org/software/make/make.html . You can check your version of gnu make by typing 'make -v'. Some vendor provided make utilities may also work, however if you find you are having problems in this area please try obtaining and using gnu make. Check that 'make' is in your path.

2.1.4 C Compiler

In general, an ANSI compliant C compiler should be used. On some hardware ANSI compliance requires a compiler flag, if you're not sure check the documentation for your compiler. Some platforms such as Solaris and HPUX do not come with an ANSI compliant C compiler by default. If you have not purchased that additional product from the vendor, we recommend GNU C (gcc) available at http://www.gnu.org/software/gcc/gcc.html. Check that the C compiler is in your path.

- With the 'pgf90' FORTRAN compiler, 'pgcc' is recommended.
- With the Intel 'ifort' FORTRAN compiler, 'icc' is recommended.
- For Solaris platforms, 'cc' is recommended.
- For HP-UX platforms, 'cc' is suggested.

2.1.5 FORTRAN Compiler

Please note that LAPS uses dynamic memory within the FORTRAN code in the form of automatic and allocatable arrays, as well as other FORTRAN 90 constructs. This implies that you will need an 'f90' compiler or the equivalent. LAPS will no longer work on most 'f77' compilers. Check that the FORTRAN compiler is in your path.

- For IBM/AIX platforms 'xlf' is recommended.

- For Solaris & HP-UX platforms, ‘f90’ works well.
- For Linux platforms (i386,i686,x86_64), ‘pgf90’ is suggested while ‘ifort’ and ‘gfortran’ are being tested.
- For Linux platforms (Alpha chip), ‘fort’ is suggested (normal serial use).

2.1.6 Disk Space

The disk space requirements for LAPS vary depending on factors such as domain size and purge parameters. As a general guide, 10MB would be needed for source code. About 30MB are needed for executable binaries. 500MB to 1GB are typically needed for 12-24 hours worth of output data. A similar amount of space is needed for the raw input data.

2.1.7 Memory Etc. (ulimit)

‘ulimit’ settings should be placed at ‘unlimited’ if possible. Memory requirements vary for LAPS. As a general guide, 128MB is needed and 256MB is preferred. More is needed for large domains. For very large domains, a rough guide to the memory needed would be 100 x NX x NY x NZ bytes.

2.1.8 Plotting / NCAR Graphics Library (optional)

Lapsplot is an optional plotting program, thus NCAR graphics is optional. If you wish to build the lapsplot process, access to NCAR graphics libraries is needed so you will be able to run the ‘ncargf77’ command in the LAPS Makefiles. You can download the free NCAR graphics (NCL) software at the URL shown below. Note that NCAR graphics libraries should be built against the same FORTRAN compiler being used in LAPS.
http://ngwww.ucar.edu
The ‘lapsplot.exe’ executable is an interactive program that reads in the netCDF LAPS files and produces a ‘gmeta’ file as output. The ‘gmeta’ file can be displayed using other NCAR graphics utilities like ‘ctrans’ and ‘idt’.
‘Lapsplot’ is designed to work with version 3.2 (or higher) of NCAR graphics. The environment variable $NCARG_ROOT should be set when configuring, compiling, or running ‘lapsplot.exe’. Before running ‘configure’, check that ‘ncargf77’ and/or ‘ncargf90’ is in your path. If you are using a compiler other than ‘f77’, check after running ‘configure’ to see that the right thing was done by inspecting ‘NCARGFC’ and ‘FC’ within ‘src/include/makefile.inc’. ‘NCARGFC’ should point to either the ‘ncargf90’ or ‘ncargf77’ script. If configure wants to use ‘ncargf90’ and you don’t yet have one, then consider making a soft link called ‘ncargf90’ that points to the ‘ncargf77’ script, or copying ‘ncargf77’ to a new location and calling it ‘ncargf90’.
If you only have an ‘ncarg90’ script (i.e. no ‘ncargf77’), you may want to also make a script called ‘ncargf77’ that lists the ‘f77’ compiler. This can help ‘configure’ do its test for making the switch over from ‘ncargf77’ to ‘ncargf90’. Lapsplot is built as a special option to ‘make’, simply type ‘make lapsplot’ or ‘make install_lapsplot’. It is not built with a plain run of ‘make’.

In order to get 'lapsplot' to compile and link properly it may be necessary to edit your own version of 'ncargf90' or even the original 'ncargf77' script. Check that the proper FORTRAN compiler, load flags, and load libraries are set in the script.
A possible alternative to fixing 'ncargf77/ncargf90' is to edit 'src/include/makefile.inc' with the full path for 'NCARGFC', and appropriate compiler for 'FC' (and possibly compiler flags) for your system (after running configure).
At times the linking of 'lapsplot' may show undefined references to library routines. This often represets a mismatch between NCAR graphics and various system libraries. Possible solutions for this include editing the library list within the 'ncargf77/ncargf90' script or switching the -Bstatic flag on or off.
'Lapsplot' can be modified to show political boundaries outside of the U.S. The following data files are relevant from the 'static/ncarg' directory:

- 'continent_minus_us.dat',
- 'state_from_counties.dat',
- 'uscounty.dat'.

These political boundary files are stored in big_endian format. These would need to be converted manually prior to using 'lapsplot', if your machine is expecting little_endian. We will consider automating this in the future.
To run lapsplot manually you can do the following:

1. setenv LAPS_DATA_ROOT to the correct path
2. run $LAPSINSTALLROOT/bin/lapsplot.exe (answer the questions it asks interactively)
3. idt gmeta

Please note that 'lapsplot' is provided to help you check out how your LAPS implementation is working. Aside from the pre-generated and "on-the-fly" web products, we do not have any other plotting or visualization packages available for distribution with LAPS at this time. Many users have interfaced LAPS with their own display software (e.g. IDV, VIS5D, AVS, IDL, NCL, NCVIEW, GEMPAK).
IDV is handy as it can read Grib-2 LAPS analysis output that can be generated via the 'laps2grib' program. Feel free to post questions about the various plotting packages to the online LAPS forum.
Another note of interest is that LAPS is visualized as an integral part of the AWIPS & LAPS systems. If you have AWIPS, then LAPS should be running on it and you can view its output on the workstation.

2.1.8.1 Web Display

The 'gmeta' file can be converted into a GIF/JPEG file for web display by using 'ctrans' in conjunction with the 'netpbm' package of image conversion programs that can be downloaded

at the link just below. Within this package our web display scripts use the 'rastopnm' and 'ppmtogif' programs.
http://netpbm.sourceforge.net/
We have the option of making pre-generated GIF images that can be displayed on the web by invoking the 'sched.pl -f dummy' command line argument. Please see section 3.2 for more info on 'sched.pl'. The associated web related scripts (for analyses) such as 'etc/www/followup_ncarg.sh' are in the repository. These "wrapper" scripts run 'lapsplot.exe' and output a set of GIF images in a directory tree that can be accessed via the web. The set of web image products are defined with configuration files in 'static/www/lapsplot.*'. Color tables are specified in 'static/www/*.lut'. Other user definable plotting parameters are located in 'static/lapsplot.nl'. A separate web related script is our "on-the-fly" page that is contained in 'etc/www/nph-laps.cgi'. This CGI/PERL script can be run via a web server. This also calls a set of scripts that wrap around 'lapsplot.exe'. The file system(s) running LAPS should be made visible on your web server. After running 'configure' the following steps will help in setting up this web page.

1. edit 'etc/www/nph-laps.cgi' and set $web_root to be the root directory of the web server (document root)

2. edit 'etc/www/laps.cgi' and set $web_root to be the root directory of the web server

3. edit 'etc/www/nph-laps.cgi' and set $ncarg_root to be the root directory of the NCL/NCAR Graphics installation

4. mkdir -p $web_root/request

5. cd $web_root/request; ln -s $LAPSINSTALLROOT/etc/www/nph-laps.cgi .

6. cd $web_root/request; ln -s $LAPSINSTALLROOT/etc/www/laps.cgi .

7. For each DOMAIN NAME (foo):

    (a) mkdir -p $web_root/domains/foo

    (b) cd $web_root/domains/foo; ln -s $LAPS_DATA_ROOT private_data

8. edit 'etc/www/nph-laps.cgi' and set $default_domain to be your favorite domain 'foo' within the domain list established in step (7)

At this point you should hopefully be able to use a web browser and run the "on-the-fly" page with something like this URL:
http://yourdomain.something/request/nph-laps.cgi

2.1.9 GRIB2 External Libraries (optional)

The background models read by lga include GRIB1 and GRIB2-formatted files.
The external compression libraries required for decoding GRIB2-formatted files are libjasper.a,

libpng.a, and libz.a. They are usually found in /usr/lib or /usr/lib64. It is recommended to have a system administrator install these external libraries if they are not already on your system. (JPEG2000 and other image compression algorithms are built into GRIB2. Library support for JPEG2000 is provided via the JasPer library. The implementation of JPEG2000 compression reduces file sizes up to 80%.)
The 'configure' script will determine if these libraries are present. If all are found, 'configure' prepares the file 'src/include/makefile.inc' with DEGRIBLIBS, DEGRIBFLAGS and CDEGRIBFLAGS values allowing the lga software to build to read both GRIB1 and GRIB2-formatted files. Without these three specific compression libraries available, lga is built to read only GRIB1-formatted files in addition to netCDF-formatted files.
There may be some occasions where the Jasper library isn't detected automatically by 'configure'. For example, if the Jasper library is placed in a location other than the system area (/usr/lib) then one can set an environment variable CPP_INCLUDE_PATH for lga to build like this: setenv CPP_INCLUDE_PATH /opt/jasper/1.900.1/include
After running 'configure', the DEGRIBLIBS value in 'makefile.inc' can be manually edited to include the path information for the Jasper library. Similarly the flags -DUSE_JPEG2000 and -DUSE_PNG can be added to the value of DEGRIBFLAGS.
The unix/linux system command 'ldd' command prints the shared library dependancies on an executable; running 'ldd lga.exe' is a helpful command in the situation when you download the LAPS precompiled binaries and need more information about shared libraries required by lga.exe.
See source directory: $LAPS_SRC_ROOT/src/lib/degrib/README_LIBS file for additional information.

2.1.10 GNUPLOT/ImageMagick for Verification (optional)

LAPS has a built in verification package and this needs installation of GNUPLOT and ImageMagick to run fully.

2.2 Installation Procedure Summary

To introduce this section, here is a hierarchical listing of some primary directories and files in the laps tree. The default LAPS structure is shown in the first tree below. These directories are created/addressed in various portions of section 2.2 and beyond.
Various "root" directories are mentioned in the form of environment variables. These can optionally be set to make it easier to follow the instructions below more literally. The installation scripts can be run without setting these variables if you'd like to enter the associated paths directly as command line input.
$LAPS_SRC_ROOT - The full path that was created when the LAPS tar file was untarred. This contains the source code and other supporting software. $LAPS_SRC_ROOT is needed for building LAPS but is not needed at runtime.

$LAPSINSTALLROOT - The full path of installed binaries and scripts (bin and etc). This is where you build the executables, configure the scripts (converted the *.pl.in to *.pl), and configure $LAPS_SRC_ROOT/src/include/makefile.inc. Note: $LAPS_SRC_ROOT and $LAPSINSTALLROOT are in many cases the same but don't have to be. $LAPSINSTALLROOT is needed at runtime.
$LAPS_DATA_ROOT - The full path to the output data and namelists. This includes lapsprd subdirectories containing both LAPS output grids and intermediate data files. $LAPS_DATA_ROOT is needed at runtime and it contains all the files configured to run an analysis domain localized to a location on earth. The $LAPSINSTALLROOT tree can drive several $LAPS_DATA_ROOTs. Input data in its "raw" form is stored outside the $LAPS_DATA_ROOT tree.
Note: $LAPS_DATA_ROOT is usually (and recommended to be) different than $LAPS_SRC_ROOT/data and $LAPSINSTALLROOT/data but they don't need to be. Also, $LAPS_SRC_ROOT/data/cdl and $LAPS_SRC_ROOT/data/static are the repository versions and should be kept pristine.
Note: the namelists you get from the tar are configured for our Colorado domain. More on localizing a domain for your own area later on.
To summarize, these three environment variables can either be part of one directory tree or split out into separate trees as further discussed at various times below.

```
/home_disk/
    raw_data/                                          (optional raw test data)
    geog/
        world_topo_30s
        albedo_ncep
        landuse_30s
    laps-m-n-o.tar
    laps-m-n-o/                               ($LAPS_SRC_ROOT=$LAPSINSTALLROOT)
        Makefile
        src/
            ingest/
        etc/                                                     (laps scripts)
        bin/                                                      (executables)
        data/    (original data tree that comes with tar file,
                  replicated and merged with templates
                  during localization)
            lapsprd/
                product_list/                                     (laps output)
            log/
            static/
                nest7grid.parms                   (namelist parameters)
                *.nl                              (namelist parameters)
                static.nest7grid                  (gridded topography)
```

```
        time/
    testdata/                          (optional, can be relocated)
        lapsprd/
            product_list/
```

In many UNIX environments, large data files are stored on a "data" disk and the source code is stored on a smaller "home" disk. Below is a typical laps directory structure for that setup. We recommend using something like this setup for most LAPS users. This type of separation makes it easier to update the LAPS source code while maintaining your data intact.

```
/home_disk/
    builds/
        laps-m-n-o.tar
        laps-m-n-o/                 ($LAPS_SRC_ROOT = $LAPSINSTALLROOT)
            Makefile
            src/                                          (source code)
                ingest/
            etc/                                         (laps scripts)
            bin/                                          (executables)
        template                              ($TEMPLATE parameters)

/data_disk/
    geog/
        world_topo_30s
        albedo_ncep
        landuse_30s
    raw_data/                               (optional raw test data)
    laps/
        data*/               ($LAPS_DATA_ROOT, set up during localization)
            lapsprd/
                product_list/                             (laps output)
            log/
            static/
                nest7grid.parms                 (namelist parameters)
                *.nl                            (namelist parameters)
                static.nest7grid                 (gridded topography)
            time/

        testdata/                      (optional, can be relocated)
            lapsprd/
                product_list/
```

2.2.1 Untarring the Source Code

Place the tar file in the directory '/home_disk' or '/home_disk/builds'. Untar the laps source code using a command like

```
prompt> gzcat laps-m-n-o.tgz | tar xf -
```

OR...

```
prompt> gunzip laps-m-n-o.tgz
prompt> tar -xf laps-m-n-o.tar
```

The $LAPS_SRC_ROOT directory will be set up one level below the tar file.
If you are having trouble running 'gunzip', the problem could be that the 'laps-m-n-o.tgz' file was corrupted during the download. In that case simply try downloading again.

2.2.2 Running Configure

Go to the $LAPS_SRC_ROOT directory and run...

```
prompt> ./configure
```

'configure' supports many options, the most important is the –prefix option which tells make where to install the laps system (FORTRAN executables, Perl Scripts, etc.). The default (if you did not use –prefix) is to install whereever the source is. The use of the –prefix option is highly recommended to make it easier to update your source code (e.g. importing a new LAPS tar file), without disturbing the binaries, data, and runtime parameters that you are working with on-site. This goes along with the second directory tree diagram shown above in section 2.
For example, to install laps in directory '/usr/local/laps' (i.e. $LAPSINSTALLROOT) use...

```
prompt> ./configure --prefix=/usr/local/laps
```

One or more data directories for running laps can be specified at runtime, if desired. A single set of binaries can thus support several data directories as described below.
Another configure option is –arch. Configure tries to get the architecture from a 'uname' command, but this can be overridden by having an $ARCH environment variable or by using –arch. The allowed values for 'arch' include 'aix', 'hpux', etc.
For more information on passing in command line flags to 'configure' run

```
prompt> ./configure --help
```

2.2.2.1 Modifying Compiler Flags

The 'configure' script automatically modifies the compiler and compilation flags by modifying 'src/include/makefile.inc' according to what type of platform you are on. Hopefully the flags will work OK on your particular platform. If you want to change the flags from the default set,

you can provide command line arguments to the 'configure' script.
Some examples based on our experience are as follows:
Solaris...

```
prompt> ./configure --cc=cc
```

For IBM/AIX platforms, you will want to override the default FORTRAN compiler with 'xlf' using the command line option –fc=xlf as follows...

```
prompt>./configure --fc=xlf
```

latforms, certain flags may be needed. '-mips3' seems to help on IRIX64 v6.2.
A second method of modifying the compiler flags is to edit 'src/include/makefile.inc', after running configure. If you find that the default compiler flags don't work for your platform or that your platform has no default, you'll need to experiment to find the right set of flags. Changes in 'src/include/makefile.inc' will automatically modify the flags used throughout laps. If you find flags that work for your platform and would like us to add them to the defaults in 'configure' please let us know via e-mail.
On Solaris for example, you may want to remove "-C" from the DBFLAGS with an edit of 'src/include/makefile.inc' to allow compiling FORTRAN debug versions of the software.
On some platforms (e.g. Linux) the linking of FORTRAN programs to netCDF and other C library routines may need adjustment. This relates to the existence and number of underscores in the C routine names when called by FORTRAN routines. Fixes for this may include a combination of changing the number of underscores in the C routines, changing the CPPFLAGS for LAPS, or changing the FFLAGS for LAPS.
As an example, with errors linking to netCDF "nf" routines, you might rebuild the netCDF C library with a different number of underscores and/or adjust the FFLAGS according to the man page in your FORTRAN compiler. On a Linux-Intel machine the netCDF library can be rebuilt with the following flags...

```
FC=pgf90
CC=gcc
CPPFLAGS=-DpgiFortran
FFLAGS=-O
```

Errors linking to other LAPS C routines can be addressed with other adjustments to the CCPFLAGS (among FORTRANUNDERSCORE and FORTRANDOUBLEUNDERSCORE) or the FFLAGS.

2.2.3 Ingest Software Changes

In this file (mainly Sec 2.3), a number of potential manual changes to ingest code are outlined prior to running 'make' and '$LAPSINSTALLROOT/etc/localize_domain.pl', especially if one is using ingest data formats other than "standard" ones used at GSD. After becoming familiar with the changes needed for your implementation, it is recommended that you develop a method to save the hand edited files in a "safe" place outside of the laps directory structure, or by using a revision control system such as CVS. This strategy would make it easier to update your implementation of LAPS with the latest 'laps-m-n-o.tgz' file from GSD, while minimizing the hassle involved with software modifications for your local implementation.

2.2.4 Running Make

The next step is to build and install the executables, this can be done by running the following (note the syntax might vary depending on the shell you are using)...

```
prompt> cd $LAPS_SRC_ROOT
prompt> make 1> make.out 2>\&1
prompt> make install 1> make_install.out 2>&1
prompt> make install_lapsplot 1> make_install_lapsplot.out 2>&1
```

Check that the executables have been placed into the '$LAPSINSTALLROOT/bin' directory. The total number should be the number of EXEDIRS in '$LAPS_SRC_ROOT/Makefile' plus 2; this includes 'lapsplot.exe'.
Lapsplot can be installed only if you have NCAR graphics.
We recommend using Gnu Make Version 3.75 or later available via ftp from any GNU site. There are many other targets within the Makefile that can be used for specialized purposes, such as cleaning things up to get a fresh start. In particular, note that a 'make distclean' is recommended before running 'configure' a second time so that things will run smoothly.

2.2.5 Geography Databases

Currently there are three mandatory geography databases required to localize a LAPS domain (with a fourth optional one). These are:

1. terrain elevation (required)
2. landuse category (required)
3. albedo climatology (required)
4. soil type [bottom/top] (optional)

The other geography data paths listed in 'static/nest7grid.parms' represent data that can be processed by the localization though are unneeded by the analyses. Hence it is unnecessary to download these and they aren't available on our software download web page.
The 30" terrain elevation data is found in the tar files for 'topo_30s'.
The landuse data is global 30" data and required to compute a land/water mask. The mask is used during localization to force consistency between the other geography data at land-water boundaries. Land fraction is derived from the landuse data using the water category, with valid

values ranging continuously between 0.0 and 1.0.

The global albedo climatology database has less resolution than either the terrain or landuse data. The albedo is approximately 8.6 minutes (0.144 degs) and was obtained from the National Center for Environmental Prediction (NCEP). This data is used in the LAPS cloud analysis with visible imagery data.

The geography data come in compressed tar files separate from the rest of the LAPS distribution. The data are used in process 'gridgen_model' which is the fortran code to process all the geography data as specified by the user (see section 3.5.4 for more information about gridgen_model). Only one copy of the geography data is required no matter how many LAPS 'dataroot' installations you are supporting. The paths to the geography data directories (topo_30s, landuse_30s, and albedo_ncep) are defined as runtime parameters within the 'nest7grid.parms' file (Sec 2.2.6). The geography data is available on the LAPS Home Web page (software link). You will find the following global data sets at this web/ftp site. Some of the data have been subdivided into "quartershperes" for easier downloading. Select the files needed for your application or get all of them if you intend to generate localizations around the entire globe. .

- 132446109 Aug 24 2001 topo_30s/topo_30s_NE.tar.gz
- 63435504 Aug 24 2001 topo_30s/topo_30s_NW.tar.gz
- 37194099 Aug 24 2001 topo_30s/topo_30s_SE.tar.gz
- 29204244 Aug 24 2001 topo_30s/topo_30s_SW.tar.gz
- 12324069 Aug 24 2001 landuse_30s/landuse_30s_NE.tar.gz
- 6118611 Aug 24 2001 landuse_30s/landuse_30s_NW.tar.gz
- 3355822 Aug 24 2001 landuse_30s/landuse_30s_SE.tar.gz
- 2808861 Aug 24 2001 landuse_30s/landuse_30s_SW.tar.gz
- albedo_ncep/A90S000E albedo_ncep/A90S000W albedo_ncep/AHEADER

We are currently working on a procedure to access higher resolution terrain and land use data from the USGS (at least to 1 arcsec).

Soil Type and Other Optional Databases:

The laps process 'gridgen_model' described below in section 4 can also process soil type, mean annual soil temperature, and greeness fraction but these are not mandatory data required in LAPS and therefore we do no describe them here. Soil Type can however be used in the soil moisture analysis. You'll see some reference to these data bases below and we have added paths to this data in our namelist file (nest7grid.parms) but you should enter dummy paths for these data in the event you do not have them available. The gridgen_model process will warn that these data are not available but you should still see the localization run to completion (ie., static.nest7grid is generated).

2.2.5.1 High Resolution Terrain (sub-kilometer)

High resolution terrain can be imported (experimentally) into LAPS via two methods, the WRF Wizard (see section 2.2.7), and Topograbber - see http://laps.noaa.gov/topograbber/
The WRF Wizard can be used to generate a GEOGRID file on the same grid as the LAPS analysis. The terrain from this file can be imported during the LAPS localization by setting the 'nest7grid.parms' namelist path parameter 'path_to_topt30s' to contain the string 'wps' in the directory name.
Also, Topograbber is under development and this might be used with some further work. The tiles produced may need some modifications to the 'gridgen_model.exe' program so they can be read in. A second way to use Topograbber with LAPS is to generate a WPS GEOGRID terrain file, and then read that in during the LAPS/STMAS localization process (see above paragraph).

2.2.6 Localizing for Single or Multiple Data Domains

Runtime parameter changes may be needed to tailor LAPS for your domain(s); this includes ingest and geography data path names, grid dimensions, grid location, and potentially other aspects of the data processing. The parameter files are 'data/static/nest7grid.parms', 'data/static/*.nl', and 'data/static/*/*.parms'.
The localization involves several operations. The parameter files are merged/updated with the repository versions if needed. The dimensions in the 'cdl' files are also adjusted. Then several executable programs are run including 'gridgen_model.exe' and 'gensfclut.exe' as per section 4.1.
Below are two mainly equivalent procedures for localizing LAPS to set up one or more domains. The first is a newer, more efficient (and highly recommended) method using domain "template" directories. The second is our original method for localization. You'll want to use either Method 1 or Method 2 but not both.

2.2.6.1 Localization Method 1

The first method is especially useful if you are using a separated data tree and/or multiple domains. It is also recommended if you are doing repeated software updates. Once you learn this method it can save a lot of time and errors that may occur in the course of using Method 2.
SETTING RUNTIME PARAMETERS
If you are working in a separated data directory (e.g. using the second tree shown above), you can set up a copy of the runtime parameter files (for each window) in a new directory (called $TEMPLATE) with a reduced parameter subset. The $TEMPLATE directory namelist files should contain only those parameters that need to be changed for each of the domain(s) from the settings in the repository, $LAPS_SRC_ROOT/data/static. The remaining unchanged parameters should be omitted from the $TEMPLATE versions. Otherwise the template namelist looks exactly like the originally supplied namelist, except that the comment section should be omitted. The modified $TEMPLATE parameters generally include map projection settings,

data paths, etc. The remaining fixed parameters will later be automatically merged in from the '$LAPS_SRC_ROOT/data/static' directory tree by the localization scripts (next step).
Templates should be maintained in a location separate from the LAPS distribution and LAPS_DATA_ROOT (e.g. see the template directory in the tree diagrams above). This avoids them being erased during software updates and relocalizations. Thus templates can be thought of as more permanent, since they contain parameters dependent on the local implementation and relatively independent of software updates. Once you set up the template directory you'll be ready to run the 'window_domain_rt.pl' script. Here is an idealized example illustrating the namelist merging process that is done during the localization...

```
       template          repository tar file            localized result
       --------          -------------------            ----------------

   $TEMPLATE/vad.nl  $LAPS_SRC_ROOT/data/static/vad.nl $LAPS_DATA_ROOT/static/vad.nl
   ................  ................................. .............................


                                a=1                           a=1
        b=5                     b=2                           b=5
                                c=3                           c=3
        d=6                     d=4                           d=6
```

And here is an example of an actual template for the 'nest7grid.parms' file...

```
&lapsparms_nl
C80_DESCRIPTION = 'NOAA/GSD LAPS running for Taiwan CWB'
C6_MAPROJ = 'lambrt',
STANDARD_LATITUDE = 10.0,
STANDARD_LATITUDE2 = 40.0,
STANDARD_LONGITUDE = +120.0,
GRID_CEN_LAT = +23.578,
GRID_CEN_LON = +120.91,
GRID_SPACING_M = 9000.,
NX_L = 153,
NY_L = 141,

path_to_topt30s='/home/data_disk/geog/world_topo_30s',
path_to_landuse30s='/home/data_disk/geog/landuse_30s',
path_to_albedo  = '/home/data_disk/geog/albedo_ncep',
path_to_raw_profiler='/pj/fsldat/point/profiler/netcdf/',
path_to_qc_acars='/home/data_disk/dat/acars/',
c8_project_common='CWB',
/
```

LOCALIZING with 'window_domain_rt.pl'

Generating new localizations, reconfiguring existing localizations, and reconfiguring existing localizations without removing lapsprd or log information is made easier with the perl script '$LAPSINSTALLROOT/etc/window_domain_rt.pl' ("window" hereafter). The window script makes use of namelist domain templates that specifically define a user's localizations. The window script uses environment variables $LAPS_SRC_ROOT, $LAPSINSTALLROOT, and $LAPS_DATA_ROOT, however, -s, -i, and -d command-line inputs override those environment variables as necessary depending on user needs. The -t command-line input specifies the domain template directory and the script saves the log/lapsprd history if command line switch '-c' is not used; or, completely removes $LAPS_DATA_ROOT, then does a 'mkdir $LAPS_DATAROOT' if '-c' is supplied. The '-w laps' is always required. The window script can be run manually when configuring or reconfiguring localizations. Window copies the domain template namelists (partial nest7grid.parms or *.nl's) into a new "static" subdirectory which, in turn, are merged with the full namelists by script localize_domain.pl. Recall that $LAPSINSTALLROOT contains bin/ and etc/ while $LAPS_SRC_ROOT contains the untarred full namelists from the repository.

In the event that $LAPS_SRC_ROOT does not exist, a data/ subdirectory containing static/ and cdl/ must be available for use by 'localize_domain.pl' (i.e. $LAPS_SRC_ROOT = $LAPSINSTALLROOT). Even though it is possible to have $LAPS_SRC_ROOT/data = $LAPSINSTALLROOT/data = $LAPS_DATA_ROOT, this is not recommended since it does not allow multiple localizations. Templates will ensure that specific namelist modifications are merged with the untarred full namelists. Templates also ensure that specifics to a localization are merged into new software ports and new namelist variables (available with new software) are merged into existing localizations.

Examples:

```
setenv LAPS_SRC_ROOT /usr/nfs/common/lapb/operational/laps
setenv LAPSINSTALLROOT /usr/nfs/lapb/operational/laps
setenv LAPS_DATA_ROOT "any new or existing LAPS_DATA_ROOT"

cd $LAPSINSTALLROOT/etc

a) perl window_domain_rt.pl -s $LAPS_SRC_ROOT -i $LAPSINSTALLROOT
            -d $LAPS_DATA_ROOT -t "full path to template directory" -w laps -c

   result: all required information is provided on the command line.
           Window will use the command line info instead of getting
           The paths from the environment.

b) perl window_domain_rt.pl -w laps:
```

```
   result: lapsprd and log saved; operational namelists and cdl's are copied
           into $LAPS_DATA_ROOT/static; $LAPSINSTALLROOT/bin/gridgen_model.exe
           runs to regenerate static.nest7grid. "Saved" lapsprd and log are
           restored into $LAPS_DATA_ROOT.

c) perl window_domain_rt.pl -c -w laps:

   result: same as b) although lapsprd and log are removed and regenerated by
           "etc/makedatadirs.pm"

d) perl window_domain_rt.pl -t "full path to template directory" -w laps

   result: similar to b) but namelist specifics are copied to $LAPS_DATA_ROOT/
           static and merged with full namelists in $LAPSINSTALLROOT.

setenv LAPS_SRC_ROOT /awips/laps
setenv LAPSINSTALLROOT /data/fxa/laps_data
setenv LAPS_DATA_ROOT /data/fxa/laps

e) perl window_domain_rt.pl -t /data/fxa/laps_template -s /awips/laps \
                       -i /awips/laps -c -w laps

   result: Specific AWIPS relocalization command for lapstools GUI when run
           within the AWIPS workstation. GUI writes user input to
           laps_template/ (subset namelists; e.g., nest7grid.parms);
           $LAPS_DATA_ROOT/static and cdl/ are moved to laps_data/;
           $LAPS_DATA_ROOT is removed; new $LAPS_DATA_ROOT is generated and
           subdirectory structure created by "etc/makedatadirs.pm";
           laps_template namelist are copied to new $LAPS_DATA_ROOT;
           localize_domain.pl merges $LAPSINSTALLROOT/ and regenerates
           static.nest7grid.
```

If you decide to manually change any parameters in $LAPS_DATA_ROOT/static after running the localization, it is suggested to make the same change in the $TEMPLATE directory as well. This will help preserve your local changes in the future if you install an updated version of LAPS.

2.2.6.2 Localization Method 2

This method is included partly for historical reasons and can be useful if you haven't yet learned how to use template directories and/or the separated $LAPS_DATA_ROOT (see method 1). This procedure provides a result equivalent to that from Localization Method #1 and provides

an alternative method (even if not recommended) of modifying the parameters.
For each domain you wish to create, run...

```
prompt> cd \$LAPSINSTALLROOT/etc
prompt> perl makedatadirs.pl --srcroot=$LAPS_SRC_ROOT --installroot=$LAPSINSTALLROOT
--dataroot=$LAPS_DATA_ROOT --system_type=laps
```

where the path name \$LAPS_DATA_ROOT must be named differently for each data domain if there is more than one. Recall that each domain can be set up in a separate subdirectory under '/data_disk/laps'. Next, follow the setup and localization steps below.
The order of the command line arguments is important, but only the first one is required. If for example a \$LAPS_DATA_ROOT is not supplied, the dataroot tree location will default to where the LAPS binaries are installed via configure. Thus, the default value of \$LAPS_DATA_ROOT is '\$LAPSINSTALLROOT/data'.
The runtime parameters should be emplaced and/or modified within each \$LAPS_DATA_ROOT directory tree prior to running the localization. More details on 'nest7grid.parms' and other parameter files are discussed in subsequent parts of section 2.
As one option you can edit the parameter files that are in '\$LAPS_SRC_ROOT/data/static' and tailor them for your domain. If you have '\$LAPS_DATA_ROOT' different from '\$LAPS_SRC_ROOT/data/static' then a good alternative may be to copy any parameter files you need to edit into '\$LAPS_DATA_ROOT/static' from '\$LAPS_SRC_ROOT/data/static'.
Finally, you can create the static data files and look up tables specific to the domain(s) you have defined in 'data/static/nest7grid.parms' and other runtime parameter files. Shown below is an example of running the localization for a particular laps domain. This should be repeated (with a unique dataroot) for each domain if there is more than one.

```
prompt> cd \$LAPSINSTALLROOT/etc
prompt> perl localize_domain.pl --srcroot=$LAPS_SRC_ROOT --install_root=$LAPSINSTALLROOT
--dataroot=$LAPS_DATA_ROOT --which_type = 'laps'
```

2.2.6.3 Localization with LAPS GUI

LAPS has a GUI interface under development that can be used to localize the domain. This can be found in the '\$LAPS_SRC_ROOT/gui' directory. There it can installed using the 'install_gui.pl' PERL script as outlined in the local README file.

2.2.7 WRF Domain Wizard LAPS Support

The WRF Domain Wizard can be used to help specify correctly navigated LAPS domain map parameters. When the Wizard is run it will write out a 'nest7grid.parms' file for each nest that can be used as input "templates" for LAPS localization.
http://www.wrfportal.org/DomainWizard.html

2.2.8 MPI Support for LAPS Wind Analysis

There is capability to compile and run the wind analysis (wind.exe) using MPI. We do this by doing a separate software build with 'mpif90' and then 'sched.pl' runs the serial versions of most things while running the parallelized version of the wind analysis. To build LAPS using 'mpif90' edit the 'makefile.inc' file, between running 'configure' and 'make', adding -DUSEMPI into the CPPFLAGS.

The 'sched' script presently submits multiple processor jobs using the SGE queueing environment. We may consider adding an option to 'sched.pl' to run/submit directly with 'mpirun' if that would be useful.

Chapter 3

Running the LAPS Ingest/Analysis Code

3.1 Raw Data Ingest

There is a layer of "raw" data ingest code that may have to be modified for the individual location depending on data formats. Its purpose is to reformat and preprocess the various types of raw data into simple common formats used by the subsequent analyses. It also helps to modularize the software.

Working with the ingest code is usually the largest task within the porting of LAPS. The supported component of the LAPS code is the analysis section. Ingest code is supported only if your raw data looks has the same configuration and format as GSD's raw data. It is the reponsibility of the LAPS user to modify the LAPS ingest code if necessary to generate the intermediate data files that are inputs to the analysis code.

A flow chart for the ingest processes can be viewed in figure 3.1 or may be found at this URL: http://laps.noaa.gov/doc/slide1_v3.gif

The default LAPS ingest code obtains “raw” data, generally from the GSD NIMBUS system. The raw data can either be in ASCII, netCDF (as point data), or netCDF (as gridded data - generally not on the LAPS grid). Note that the ingest code is also generally compatable with raw SBN/NOAAPORT data as stored in netCDF files on the WFO-Advanced system. The ingest code processes the raw data and outputs the LAPS “intermediate” data files. The intermediate files are generally in ASCII for point data and netCDF format for gridded data that have now been remapped onto the LAPS grid. Most ingest code is located under the ‘src/ingest directory’. When netCDf format is used for the raw data, a cdl file for the raw data is sometimes included in the source code directory.

Depending on the data source, you may generally prefer one of three choices:

1. Convert your raw data to appropriate netCDF formats then run the LAPS ingest code as is. The CDLs and sample raw/NIMBUS netCDF files supplied with our test dataset can serve as a guide to writing the software to do this. If the CDL is unavailable, doing ‘ncdump -h’ on the actual data file will yield equivalent information. We generally do not maintain or support any software for writing “raw” netCDF files as this is done external to LAPS. Sometimes by posting a message to ‘laps-users’ you can obtain information from other LAPS users as to how they may have implemented this step.

2. Run a process independent of the LAPS ingest code that creates the intermedate data file.

3. Modify LAPS ingest code to accept your own raw data format. This often entails writing a subroutine that reads the data and linking this routine into the existing ingest process. That process then writes out the LAPS intermediate file. Note that generating an GSD style raw data file is not here needed - all that really counts is producing an intermediate data file. Recommended only for advanced users or those who believe their modifications have enough general interest for inclusion in the baseline LAPS repository.

For the model background and in-situ observations generally (1) is the best option. For gridded data (satellite or radar) options (1) or (2) usually work best. Most external users should avoid option (3) unless it is done in close consultation with LAPS staff. A key consideration is how easy it will be to update your version of LAPS and have it work with your local data.

You may note the following data sources used at GSD. These data sources are what the GSD ingest code is tailored to for producing intermediate data files. Note that LAPS will still run even if some of the data sources are withheld, albeit in degraded fashion. A minimum dataset of model background and surface observations is generally needed to get reasonable results.

The pathnames for the ingest data sources are assigned within the ‘./data/static/nest7grid.parms’ and other ‘*.nl’ files and can be set accordingly at runtime. Doing a grep for ‘path’ in these files will give you a quick listing of the relevant parameters.

Unless otherwise specified, the time window for data in the intermediate data files should be ‘+/- laps_cycle_time’. The time window for data in the raw data files is more variable and is generally specified within the raw data (e.g. in the CDL).

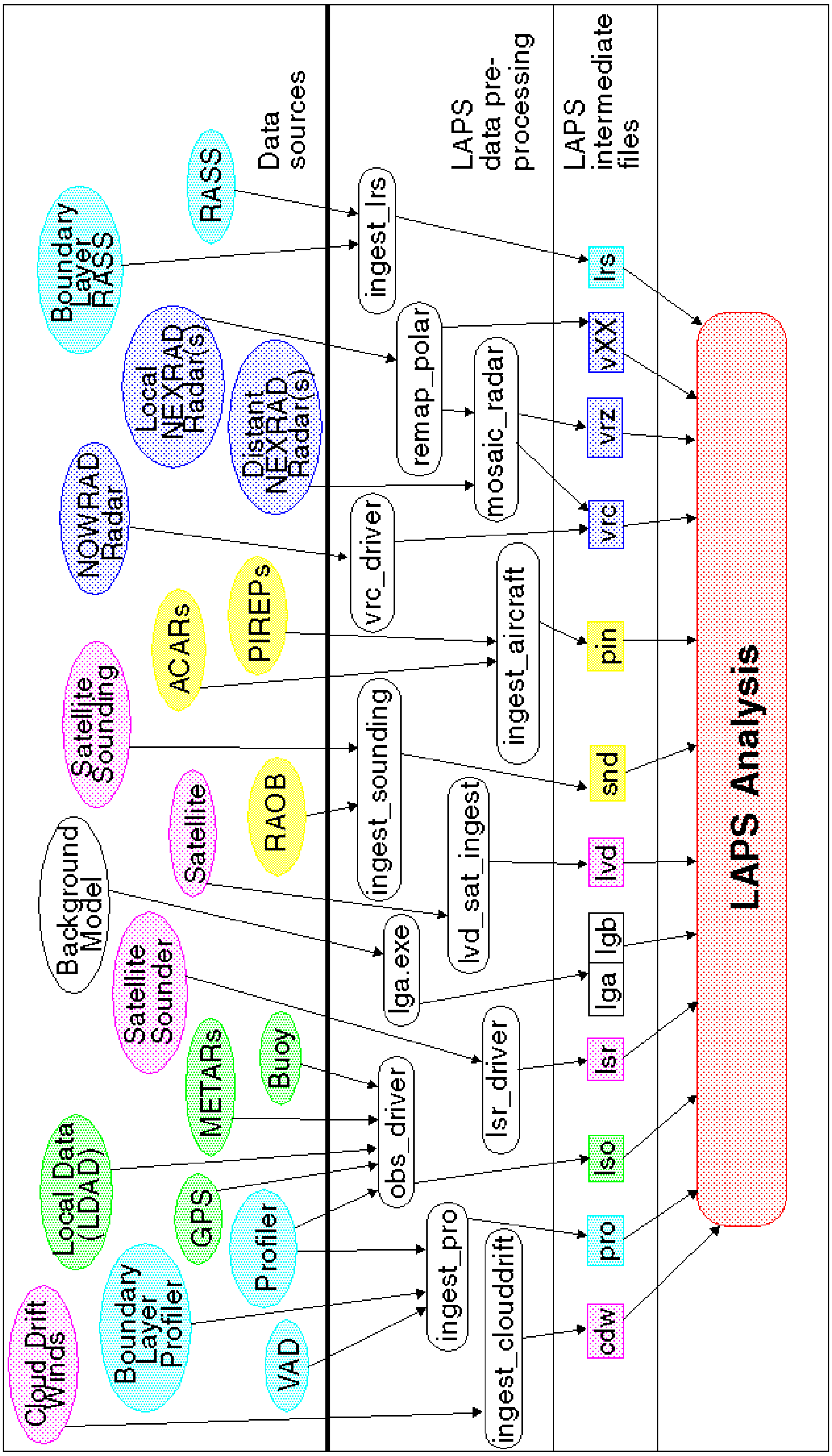

Figure 3.1: Flow chart for the ingest processes

Further information on specific LAPS ingest processes for the various data sources is found in chapter 4 of this README.

3.1.1 Model Background (lga/lgb)

The model first guess (background) is generally on a larger-scale grid than LAPS and is run independently. The model data is interpolated to the LAPS grid by the LAPS ingest to produce 'lga/lgb' files. This 'lga/lgb' output is distinct from the 'fua/fsf' files that are first guess files of similar format generated by the LAPS forecast model using an intermittent 4dda mode.

The 'nest7grid.parms' namelist variable "fdda_model_source" controls the background used in the analysis, including lga. A list of "fdda" backgrounds that are available with this release are specified in file etc/laps_tools.pm - module mkdatadirs. Even though fdda subdirectories are populated with current backgrounds, the analysis can be forced to override this by making the first entry of "fdda_model_source" = 'lga'.

The acceptable models and formats for the background model are listed in 'data/static/background.nl'. Many models can be accepted in netCDF format. A new capability in LAPS is to process GRIB input without first converting to netCDF format. For Grib data to be decoded an associated Vtable.XXX needs to be found in directory 'data/static/Variable_Tables'. The Vtable can be configured for either GRIB-1 or GRIB-2. However we are unable to guarentee that any model specified in 'background.nl' will work without some software modification.

Rapid Refresh (RR) grids are ftp'ed from NCEP to GSD, then converted at GSD from GRIB to netCDF. This netCDF file is the input for the LAPS ingest process that writes "lga". For more information on RR check the following URL for more info: http://rapidrefresh.noaa.gov/ Note that we often read these into LAPS as RUC (Rapid Update Cycle) look alike files.

RR is also available from UNIDATA and distributed to universities through private companies like Alden.

The conversion from GRIB to netCDF is done outside of LAPS by GSD's Information and Technology Services (ITS) group (in the NIMBUS system). Having the CDL should mostly be sufficient along with general knowledge of netCDF for writing out the data. Beyond that, you may wish to contact the ITS group for more info (see the reference to them in section 4.2.1). The Atlanta, Sterling, and Seattle WFOs have followed a more direct route, going from the RR/Eta to the intermediate "lga" file, bypassing the netCDF file on the model grid. This includes RR on isobaric surfaces.

3.1.1.1 Acquiring Model Background Data

GRIB-formatted background model files are now supported and can be directly read into lga.

- Where Can Users Find GRIB Data? At the NCEP ftp server for real time data sets located at this ftp site. These products can be downloaded from the web or via anonymous ftp. The following is a discussion for locating and acquiring NAM, GFS, and RUC model backgrounds for use with lga. The models are available in grib1 and grib2 formats as indicated.

- NAM Model: NAM 221 High Resolution North American grid, 32-km can be found at this ftp site with the directory and filenames as follows

  nam.{YYYYMMDD}/nam.t{CC}.awip32{FF}.tm00{.grib2}

  where YYYYMMDD is the current date, CC is the model cycle time (00, 06, 12, or 18) and FF is the forecast hour (00-84). awip32 indicates the 32 km North America (NCEP grid 221).
- GFS Models: GFS global longitude-latitude grid (360x181) 1.0 deg (fh 00-180) can be found at this ftp site adding

  {YYYYMMDDHH}{CC}/gfs.t{CC}z.pgrbf{XXX}{.grib2}

  as suffix, and GFS global longitude-latitude grid (720x361) 0.5 deg (fh 00-180) can be found at this ftp site adding

  gfs.{YYYYMMDDHH}{CC}/gfs.t{CC}z.pgrb2f{XXX}

  as suffix, where CC is the model cycle time (i.e. 00, 06, 12, 18) and XXX is the forecast hour of product from 00 - 180.

  The 1.0 degree GFS uses file identifier 'pgrb' (pressure-based grib) and is now available in grib2 as well when '.grib2' is present. The 0.5 degree GFS uses 'pgrb2' (pressure-based) and is only available in grib2.
- RUC Model: RUC Rapid Update Cycle 40km and 20km pressure data sets can be found at this ftp site adding

  ruc2a.{YYYYMMDDHH}/ruc2.t{CC}z.pgrb{XXX}{.grib2}

  as suffix where CC is the model cycle time (i.e. 00, 06, 12, 18) and XXX is the forecast hour of product from 00 - 12 (or more). File identifier 'pgrb' is used for the 40km resolution and 'pgrb20' is used for the 20km.

Additional description of NCEP products can be found at this website. A master list of NCEP GRIDS ID numbers (e.g. 211) and other specifications can be found at this website.

- How Do Users Name The GRIB Data Files? For LAPS ingest at NOAA/ESRL, we have a process that automatically downloads GRIB files to a designated directory. For example,
  - '/data/grid/gfs/global_0p5deg/',
  - '/data/grid/gfs/global_1p0deg/',
  - '/data/grid/gfs/conus211/',

  are three directories for the GFS global 0.5 degree, global 1.0 degree and CONUS 211 domains. The files within these directories are renamed from the complex patterns listed above to filenames with the following pattern: 'YYJJJHHMMhhhh'. Here the 'hhhh' part represents the number of hours into the forecast. Thus a file for GFS CONUS

211 initialized on Jul 23 2008 at 1200 UTC, with a 6 hour forecast would be named '/data/grid/gfs/conus211/grib/0820512000006'. The HRRR model follows a slightly different convention of 'YYJJJHHMMhhmm', so that forecasts of under one hour can be represented.

- How Does lga.exe Know Where To Find The Data? For lga.exe, the acceptable models, directory paths and file formats are identified in 'data/static/background.nl'. In the example above if we wanted to use the US-scaled data, we would set bgpath='/data/grid/gfs/conus211/grib/', bgmodel=13 (for GRIB), and cmodel='GFS'.

3.1.2 Radar Ingest

The following are intermediate files for various forms of radar data. These may have already been pre-processed (remapped) from "raw" data, and at this stage are in Cartesian format on the LAPS grid.

```
(vrc)                  - Low-level reflectivity from single or
                         multiple radars. For example, our ingest
                         at GSD processes WSI-NOWRAD, stored at GSD
                         in netCDF, to create the 'vrc' intermediate
                         file. Narrowband single-tilt data from AWIPS
                         is also stored in 'vrc' files.

(v01, v02, ... )       - WSR-88Ds or other radar data are stored as
                         a full volume.

                       - Each 'vxx' file has 3-d reflectivity,
                         velocity, and nyquist velocity for one
                         radar. Horizontal and vertical gaps are
                         filled in for reflectivity while sparse
                         arrays are used for velocity.

(vrz)                  - 3-D reflectivity mosaic from
                         multiple radars ('vxx' files).

(ln3)                  - Layer reflectivity and echo tops, from a
                         single radar or mosaiced from multiple
                         radars. For example, WSI sends put a variety
                         of derived products from the WSR-88D's
                         which we call nexrad products. These
                         include 3 layer reflectivity products,
                         a composite reflectivity, echo tops, and
```

```
vil.  FD also decodes these and writes
netCDF files. We have an experimental
process called ‘ln3’ that ingests these
data that we haven’t yet approved for
release in our tar. This will probably
happen shortly though as the need
to further test/use this 3D reflectivity in
LAPS is increasing.
As of 9-22-98 we have committed the ln3 ingest
process to our repository and distribute this
source code with our tarfile. The reliability
of echo tops in conjunction with the layer
reflecitivity information still makes this a
problematical data set to use in the analyses.
Committing this source to the repository will
help to further investigate the utility of this
product.
The key fields from ‘ln3’ which are used
in the analyses are the Layer reflectivity
(0-4 km MSL, 4-8 km MSL, and >8 km MSL), as
well as echo tops (MSL).
```

A flow chart showing radar data usage in LAPS is on the Web at this website, together with some text details at this website. These include information on which types of radar data are processed via the various intermediate data files.

Further information on using individual radar ingest processes is in section 4. Specifically we should establish whether your raw data is in polar or Cartesian form. If polar, please take a look at "Polar Radar Data" in section 4.2.3. NOWRAD / WSI (Cartesian) data is covered separately within section 4.2.4.

3.1.3 Surface Data

Sfc Obs (lso): GSD uses surface observations as input with the default being GSD's NIMBUS netCDF format.

Surface observations of various types, covering much of the world are available in realtime from GSD's MADIS system (with some restrictions). This data, generally in 'WFO/AWIPS' netCDF format, are distributed via the MADIS server at http://www-sdd.fsl.noaa.gov/MADIS/. The supported MADIS surface observation datasets include 'metar' (METAR/SYNOP), 'maritime' (Buoy/Ship), 'mesonet', 'urbanet', and others. This is an excellent source of surface observations for most users outside of ESRL/GSD to start with. To request a real-time data stream please go to the MADIS data application page at this link:
http://www-sdd.fsl.noaa.gov/MADIS/data_application.html

A few other METAR/SYNOP formats are now being supported in LAPS software as listed in the 'static/obs_driver.nl' namelist. The GSD code is in the 'src/ingest/sao' directory, and includes routines to read and reformat various surface data types (METAR/SYNOP, mesonet or local/LDAD, buoy/ship or maritime, and GPS/profiler surface obs). There is a subroutine tree outline in the 'src/ingest/sao/README' file including information on the supported data formats for each observation type. Paths to the datasets are specified in the 'obs_driver.nl' file

In most cases users should be able to convert their surface observations into the 'NIMBUS' or 'MADIS' NetCDF formats.

Note that only observations reasonably near the standard shelter height (2 meters, except 10 meters for wind) should be included in the LSO file. Tower mounted instruments should instead be placed in the SND file using "TOWER" for the observation type.

3.1.4 Wind Profiler/RASS

Profilers/RASS (pro/lrs) - The raw data are obtained from GSD's NIMBUS database and/or AWIPS in netCDF format where they are stored in four different directories. The data originally come from GSD's Demonstration Branch (DB) from two main networks. The 30+ NPN (National Profiler Network - NOAAnet) profiler network is located mostly in the central U.S.

The second network supplies boundary layer profilers for both wind and temperature, with formats including NIMBUS, MADIS Multi-Agency Profilers (LDAD), and RSA (LDAD) format.

The profiler data for wind goes into the 'pro' intermediate file, and RASS temperature profiles go into the 'lrs' intermediate file. Note that the cdl's associated with each data source indicate the time frequency of the data that our ingest code can process. par To summarize...

```
Network      Database(s)          file   frequency     cdl(s)
-------      -----------          ----   ---------   -----------
NPN wind     NIMBUS/AWIPS/MADIS/  PRO     404 MHz      wpdn60.cdl
                                                       wpdn06.cdl

NPN RASS     NIMBUS               LRS                  rass60.cdl
                                                       rass06.cdl

BLP wind     NIMBUS               PRO     915 MHz      wpdn60.cdl
                                                       wpdn30.cdl*

BLP RASS     NIMBUS               LRS                  rass60.cdl
                                                       rass30.cdl*

BLP wind     MADIS-MAP/RSA        PRO     915 MHz
                                           50 MHz
                                           SODAR
```

```
BLP RASS     MADIS-MAP/RSA        LRS     915 MHz
                                           50 MHz

* Indicates that the data with this cdl is available, but our
  ingest code would need modification to process it.
```

The NPN wind profiler data is available via another route from GSD with some restrictions. This data, in 'WFO/AWIPS' netCDF format, is distributed via GSD's MADIS project at http://www-sdd.fsl.noaa.gov/MADIS/

3.1.5 PIREPS & ACARS from aircraft

PIREPS (pin) - We are ingesting GSD NIMBUS and WFO/AWIPS (netCDF) pirep files to translate the cloud layers from voice pilot reports into intermediate "PIN" files.

ACARS (pin) - We are ingesting GSD NIMBUS, WFO/AWIPS (netCDF) and AFWA databases for ACARS data to translate the automated aircraft observations. The wind, temperature and humidity obs are appended to our intermediate "PIN" file. A NIMBUS equivalent netCDF database is available (with some restrictions) on the Web via MADIS at http://www-sdd.fsl.noaa.gov/MADIS/

Note the TAMDAR is presently being screened out from the NIMBUS database while this data source is being validated.

3.1.6 RAOB/Dropsonde/Radiometer

RAOBs (snd): GSD NIMBUS, WFO/AWIPS, CWB, or AFWA databases. These are available in real-time from GSD with some restrictions. RAOB data in 'WFO/AWIPS' netCDF format is distributed via GSD's MADIS project at http://www-sdd.fsl.noaa.gov/MADIS/

Dropsondes (snd): A Dropsonde ingest module has been developed for the CWB database. An ingest module has also been developed for AVAPS. We now allow the SND format to be used as input (so far just for the "AIRDROP" project). For the SND input option, the ingest program simply does a time windowing of the raw data. We may include modules for other (e.g. netCDF) databases in the future, such as NIMBUS or WFO/AWIPS.

Radiometers (snd): A radiometer ingest module has been developed for the MADIS database - http://www-sdd.fsl.noaa.gov/MADIS/

3.1.7 Satellite

Satellite Image Ingest (lvd): GOES data ingest. Data is acquired at GSD's ground station and stored in netCDF. We also obtain AWIPS/NOAAPORT/SBN data (stored in netCDF). Ingest of Air Force Weather Agency (AFWA) satellite data is also possible.

Raw GVAR satellite data can be ingested and navigated using GIMLOC routines. These files are in NetCDF. Further details can be found in the file 'src/ingest/satellite/lvd/README'.

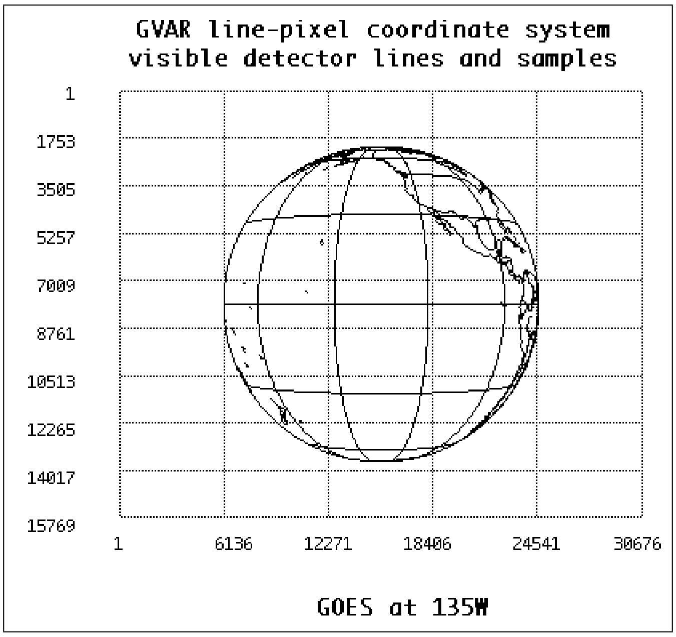

Figure 3.2: GVAR line-pixel coordinate system

The ITS group at ESRL/GSD has put together a converter from McIDAS AREA files to the GVAR netCDF format (lvd input). These files are similar to the "raw" GVAR, except they have lat/lon arrays added to make the files self navigating. The AREA files can be obtained from sources such as the NESDIS ADDE server. The Java based converter package can be found online at this URL:

http://laps.noaa.gov/software/sat/McArea2NetCDF.tar

Some tweaking of satellite coordinate and image dimensions may be needed when setting up the McIDAS package, as can be seen in the sample illustration (link below or figure 3.2). Programs like 'ncview' can be helpful to check if the window is navigated properly in the GVAR netCDF files prior to running the LAPS satellite ingest.

http://goes.gsfc.nasa.gov/pub/goes/GVAR.135W.gif

Work has been done to ingest Meteosat Second Generation data into LAPS as can be seen here:

http://www.isac.cnr.it/ meteosat/pub.html

Another option under development is to use flat files (ascii files generated by RAMSDIS or binary data) as input. The flat file ingest was still under development as of 3-11-98.

Generally it is best to convert your data into either GVAR NetCDF or remap it to create the intermediate LVD files.

Satellite Sounder Ingest (lsr): GOES satellite sounder data ingest. Program lsr_driver.exe processes data from both satellites. Product files are yyjjjhhmm.lsr and stored in subdirectories lapsprd/lsr/'satid'. Nineteen channels. Output is Radiance. The namelist 'data/static/sat_sounder.nl' defines the appropriate parameters for this ingest process. Only the moisture analysis is using this product. Currently GSD /public sounder files in netCDF format are processed. This data is useful only when GOES Vapor (GVAP) is unavailable.

Satellite derived soundings (snd): We have interfaces to GOES binary and MADIS POES (Polar Orbiter) formats. AFWA database format was previously used at GSD though not currently. The output represents derived profiles of temperature and moisture. For other formats you may wish to supply your own routine to convert your raw data into the 'snd' format.

Cloud Drift Winds (cdw): We are ingesting the ASCII satellite cloud-drift wind files for use in the wind analysis. These come from NESDIS (via NIMBUS) as well as from CWB and AFWA. We can also utilize netCDF files from MADIS. Both NESDIS and MADIS files are included in our sample data set.

3.1.8 GPS

GPS: LAPS uses GPS data from NIMBUS netCDF files. The precipitable water is used in the humidity analysis. STMAS is being designed to use the signal delay directly instead of the PW. The netCDF files are available online at ftp://gpsftp.fsl.noaa.gov where they are named according to 'GPSIPW_CDF_YYDDDHHMM0030o.nc'. The leading 'GPSIPW_CDF_' of the names would have to be stripped off to be used in LAPS/STMAS.

There are plans to make files similar to the NIMBUS ones available in AWIPS-II, though again filenaming conventions may need to be addressed.

MADIS (LDAD) mesonet files also carry the GPS PW and related data, including surface obs. However there may be some questions about the latency of this data feed for GPS. Based on tests conducted in 2011, with a LAPS cycle that begins at about 20min past the top of the hour, one can generally expect only 5-10% of the GPS data to be avaliable via MADIS when the code is configured to seek the data.

3.1.9 Other Data Sources

Radar VAD Algorithm winds (pro) GSD NIMBUS netCDF database, from WSR-88D algorithm output. GSD obtains this from NCEP and does not presently redistribute it.

SODAR data (pro) - This is treated in a similar manner to wind profilers and can be processed by LAPS ingest to appear in the PRO file. This is available as part of the RSA project at Kennedy and Vandenberg Space Centers and comes into netCDF format via AWIPS/LDAD.

Met Tower data (snd) - This is treated in a similar manner to RAOBs and can be processed by LAPS ingest to appear in the SND file. This is available as part of the RSA project at Kennedy and Vandenberg Space Centers and comes into netCDF format via AWIPS/LDAD.

Radiometric Profiler (snd): We have an interface to radiometric profilers (in netCDF via NIMBUS) that can be used for the temperature and humidity analyses.

Lightning Data: Although the LAPS repository doesn't yet have any lightning data ingest it is being considered to do this in terms of a simulated 2-D reflectivity that is one of the components of the VRC intermediate file.

3.2 Running LAPS Analyses

LAPS runs in real-time under cron; there is a sample cron script in '$LAPSINSTALLROOT/util/cronfile'. Referring to this cron, you can see that once each hour (or other cycle time), the main './etc/sched.pl' runs. As an example at ESRL, we run the 'sched.pl' hourly at :20 after the top of the hour. By inspecting the 'sched.pl' file you can see the various executables that are run in a certain order. Various command line arguments are documented within 'sched.pl' (such as '-d 0.25' that is useful for a 15-min cycle when the latency is 20 minutes). You might want to modify the 'sched.pl' file for your needs.

In the sample cron script several ingest processes are run separately from the 'sched.pl'. For example the satellite ingest (lvd) is run several times per hour and utilizes './etc/laps_driver.pl'. NOWRAD Radar ingest (vrc) is also run at more frequent intervals. You might also choose to run 'remap_polar_netcdf.exe' for radar ingest in this manner.

On many unix systems jobs that run in cron do not have access to the environment defined by the user. They instead use a system default environment defined in '/etc/profile'; thus 'perl' may not be in the $PATH. The cron file uses the full path to 'perl' to ensure that this will not be a problem. If the path to 'ncgen' is not in '/etc/profile', then you may want to add this to your own '.profile' file.

Each script in the cron requires the path to laps as a command line argument. A second optional argument specifies the path to the laps data directory structure; this path defaults to '/fullpathto/laps/data' if not provided.

The 'util/cronfile' is created by the configure step. Much of the needed editing has already been done in the creation of this file. You might see some remaining '@....@' constructs though that can be edited either manually or by running the 'cronfile.pl' (next paragraph). The @laps_data_root@ can be replaced with your path to $LAPS_DATA_ROOT and the optional @followup@ can be replaced with anything you wish to run after the 'sched.pl' has completed (using a semicolon to separate the two commands).

There is also a script called 'etc/cronfile.pl' that creates a modified version of 'util/cronfile' tailored to a given domain. This script can be run manually and the output location of the cronfile is located in '$LAPS_DATA_ROOT/cronfile'.

3.2.1 Cron timing considerations

The frequency of the cron entries for running 'sched.pl' is defined to be the LAPS cycle time. This should correspond to the value of the 'laps_cycle_time' parameter within the 'nest7grid.parms' file.

The best timing of the cron is often related to the arrival time of the raw surface observations. For example, if most of the surface data arrives within 20 minutes of the observation time, then running the cron 20 minutes after the 'systime' would be optimum. The time window for acceptance of surface stations in the LSO file can be controlled by runtime parameters in 'obs_driver.nl'.

Once we get the 'sfc_qc.exe' Kalman module operating in LAPS, we may be able to recommend running LAPS earlier. In that future mode, LAPS would process each most recent observation available and project ahead if needed any observations that did not yet arrive for the current 'systime'.

In most cases, the data cutoff time window for 3D observations is +/-laps_cycle_time/2 or +/-laps_cycle_time. For example an hourly LAPS cycle accepts RAOB data from a +/-60 minute time window and ACARS from a +/-30 minute window.

3.2.2 Purging Output Files

The script '/etc/purger.pl' purges the 'lapsprd' output files and is in turn run by the 'sched.pl'. There are default settings in place for the number of files and age of files to be kept. These can be overridden in three ways.

- The 'sched.pl' command line options '-r -m N', where "N" is the (default) maximum number of files to be kept in each product directory by the purger

- Overrides can be read in from '/data/static/purger.dat'. You can see the 'purger.pl' script to see how that information is used.

- Simply edit the 'purger.pl' to change it accordingly.

3.2.3 STMAS and other configurations

Within the LAPS cron the call to 'sched.pl' can have some optional command line arguments that adjust the runtime options. The default is to run both surface and 3-D analyses from the "traditional" version of LAPS. Here are examples of some other alternatives:

- STMAS-2D surface analysis only

  ```
  prompt> perl sched.pl -M stmas_mg.x [other regular options]
  ```

  Note that when running STMAS-2D analyses, the 'lgb_only' parameter in the 'background.nl' namelist can be set to .true. for improved runtime efficiency.

- "traditional" LAPS surface analysis only

  ```
  prompt> perl sched.pl -M laps_sfc.x [other regular options]
  ```

- STMAS-3D

  ```
  prompt> perl sched.pl -V STMAS3D [other regular options]
  ```

3.3 Test Data Case

Tar files containing test data (called 'lapsdata*') are available that contain a snapshot of several hour's worth of laps data from the Colorado domain using namelist settings taken from the repository. The tar files include intermediate files from the 'ingest' code plus outputs from the 'analysis' code. Several consecutive analysis cycles are posted with one file per cycle. Included are the contents of the 'lapsprd', 'time', 'static', and 'log' subdirectories under 'data' or $LAPS_DATA_ROOT. The log files are useful for diagnosing any differences in output you may observe. The contents of the various directories are outlined elsewhere in this README file. The data was created using the latest software release. Our users can download this data at this URL:

http://laps.noaa.gov/cgi/LAPS_SOFTWARE.cgi.

It is suggested here to test the localization procedure to ensure that all the static files needed to run LAPS are present. To do this, check that the paths to the geography data are correct in '$TEMPLATE/nest7grid.parms' and/or '$LAPS_DATA_ROOT/static/nest7grid.parms'.

When running LAPS as a whole for the archived data, the 'etc/sched.pl' script will accept a '-A' command line argument. This forces the script to run for the time you are inputting instead of the current time. An example call is shown as follows

```
prompt> perl sched.pl -A dd-mmm-yyyy-hhmm $LAPSINSTALLROOT $LAPS_DATA_ROOT
```

. . . where the inputted 'dd-mmm-yyyy-hhmm' value is the date (for example 28-Aug-2007-1500). This date can be inferred from the contents of '$LAPS_DATA_ROOT/time/systime.dat'. Best results are obtained when using a time just prior to the latest raw data tarfile time.

One can also initiate individual executables (bin directory) listed in the 'sched.pl' to run on the test data. This often helps in getting a better match between your output and ours. Note that $LAPS_DATA_ROOT needs to be set as an environment variable when executables are run individually. The time of the run is specified in '$LAPS_DATA_ROOT/time/systime.dat'. This can be modified if needed if you want to try a slightly different time from the one supplied. To do this, interactively run the script '$LAPSINSTALLROOT/etc/systime.pl' and write the standard output to '$LAPS_DATA_ROOT/time/systime.dat'.

Note that for any given process or set of processes, deviations from the GSD output may be caused by differences in the inputs as well as machine roundoff error. Most, but perhaps not all of the input data is supplied. One main area to check would be differences in available "raw" background data files. Having all of the data history from 'lapsprd' may also be an issue; this may be less of a problem if you run laps for the latest hour of data that is supplied. The history is then supplied from earlier 'lapsdata*/lapsprd' output. Output differences can be tracked down by recompiling specific analyses with the '-g' option. This can be done by typing 'make debug' in the appropriate 'src' directories. Various debuggers can then be used such as 'dbx'. Examination of the log files again is helpful.

We have a new script (in 2004) called 'casererun.pl' that can be used for archive data runs. We have yet to try it on the supplied test data case though it could prove to be useful.

3.3.1 Analysis Only Test

You may want to check that any analysis outputs from this time are not present, leaving only the 'ingest' outputs in place. This may improve the results of comparisons of your own output with GSD analysis output, though this step is not always necessary. You might consider adding the '-T' command line option when you run 'sched.pl' so that we run the analysis executables only thus skipping the ingest processes. This can be done if the ingest outputs (i.e. analysis inputs) are already present in the various 'lapsprd' subdirectories.

One way to supply the analysis inputs is as follows for each input (taken from a list of ingest outputs, see section 4.2):

```
prompt> cp testdata/lapsprd/[inputlist]/* $LAPS_DATA_ROOT/lapsprd/[inputlist]
```

OR

```
prompt> cd $LAPS_DATA_ROOT/lapsprd \\
prompt> ln -s testdata/lapsprd/[inputlist] .
```

3.3.2 Ingest + Analysis Test

For this type of test, you will want to download the 'rawdata*' tar files into your 'raw_data' directory to start the processing of LAPS. Recall that the 'raw_data' directory is on a separate tree than $LAPS_DATA_ROOT.

Raw data formats and filename conventions are consistent with the default namelist settings taken from the repository. This is generally in NIMBUS (self describing netCDF) format with associated file naming conventions. A typical filename on NIMBUS looks like this: '0606701000100o' meaning 'yydddhhmmHHMM' where 'ddd' is the day of year, 'hhmm' is the time of day and 'HHMM' is the file recurrence interval. The 'o' at the end means that observations are binned into files according to observation time (instead of 'r' for receipt time). More about NIMBUS is detailed in publications on the web at this URL:

http://www.fsl.noaa.gov/its/papers/jb_ams94.html

Note that with the RUC grib data there are two directories. The one with soft links (and without the ".grib" at the end of the filenames) is the one to use.

Time information will be needed in the form of 'data/time/systime.dat'; this can be extracted from the 'lapsdata*' tar file.

The 'raw_data' directory is a convenient place to store test data. User supplied raw data for operational runs can be stored anywhere on your system, often outside of the LAPS trees.

Note that the 'lapsdata*' tar files contain intermediate plus analysis output files only. The 'rawdata*' tar files supply much of the "raw" data that are inputted to the ingest processes. The times for the raw data match the 'lapsdata*' output approximately but not always exactly (one example being the raw background data files). As a hint with the background data check that the available raw files bracket the systime of interest. If needed one can change the 'use_analysis' flag in 'background.nl' to get 'lga.exe' to work better.

In many cases, a user could independently generate the intermediate data files (ingest output) and then compare them with ours. If other "raw" files are needed as they appear on GSD's NIMBUS & MADIS systems, please let us know and we can try to add them to our test data case or send them separately.

3.4 I/O of LAPS Gridded Files

Once the laps library is compiled (as outlined above), laps grids can be read. There are three levels of software that can access the data.

```
1. Lowest Level - netCDF c routine calls (not recommended unless you're
                                           a netCDF hacker)


2. Medium Level - READ_LAPS_DATA - look at the source code in
   lib/readlapsdata.f for the arguments.


3. Highest (and easiest) level - get_laps_3d or get_laps_2d. The source
   is contained in src/lib/laps_io.f. The various grids available are
   listed later in this README file under the heading "NetCDF organization"
```

To link to the reading routines, you will want to link to: laps/src/lib/liblaps.a libnetcdf.a

3.5 Changing the Horizontal Domain

Laps will allow you to change the horizontal domain after compilation and before the running of the localization scripts. Below is a list of the relevant changes.
The dimensions and location of the horizontal domain can be changed at run time. Prior to running 'window_domain_rt.pl', set the following parameters in 'data/static/nest7grid.parms' or in the corresponding template directory (needed only if you are outside the default Colorado domain). This script in turn runs 'gridgen_model.exe' and other programs.

3.5.1 Number of Grid Points

Adjust the horizontal dimensions in terms of the number of grid points (NX_L, NY_L) in './data/static/nest7grid.parms'.

NOTE: Various files in the ./data/cdl directory are automatically edited by ./etc/localize_domain.pl using the values found in './data/static/nest7grid.parms'.

3.5.2 Location of Domain (Map Projections)

```
1) Modify the 'grid_spacing_m' parameter (only if you want to
        change from the default 10000m for the grid spacing).


        Grid spacing in meters on the projection plane. Used for all
```

```
        projections.

2) Modify the 'grid_cen_lat' and 'grid_cen_lon' parameters.

        These are the latitude and longitude of the center of the
        domain, expressed in degrees. The parameters are needed for
        all projections.


3) c6_maproj:

        Polar stereographic:
            Set to 'plrstr'.

        Lambert Conformal:
            Set to 'lambrt'.

        Mercator:
            Set to 'merctr'.

        In most cases, the Lambert projection is recommended.

        Mercator is recommended if the domain includes the
        equator, or for domains centered in the tropics where
        sin(latitude) varies by more than a factor of two over the
        domain.

        If the domain includes one of the geographic poles, then
        Polar Stereographic should be used instead.

        See the note below regarding current map projection limitations.

4) standard_longitude:

        Polar Stereographic:
            This defines the longitude which is straight up and down
            (parallel to the "y" axis) in the map projection.

        Lambert Conformal:
            This defines the longitude which is straight up and down
            (parallel to the "y" axis) in the map projection.
```

```
    Mercator:
        N/A

5) standard_latitude:

    Polar Stereographic:
        This is the latitude at which the grid spacing is exactly
        the nominal value (‘grid_spacing_m’ e.g. 10km).

        This parameter is usually set to +/-90 degrees to match
        the latitude of the projection pole (‘standard_latitude2’),
        given that the projection pole is at one of earth’s
        geographic poles. The actual grid spacing (measured on
        the earth’s surface) matches the ‘grid_spacing_m’
        parameter at the projection pole, which may or may not be
        located within your domain. For domains distant from the
        projection pole, the actual grid spacing inside the domain
        becomes noticeably less. The value of ‘grid_spacing_m’
        can be increased to compensate. The projection plane is
        tangent to the earth’s surface.

        When the projection pole is at a geographic pole,
        ‘standard_latitude’ can be set to values other than +/-90.
        The ‘grid_spacing_m’ parameter then represents the
        true grid spacing (measured on the earth’s surface) at a
        latitude of ‘standard_latitude’. The projection plane
        is secant to the earth’s surface.

        Consider the angle ‘psi’ which is the angular distance from
        the pole of the projection. ‘phi’ = 90 - ‘psi’. The map
        factor ‘sigma’ is (1+sin(phi0))/(1+sin(phi)) and becomes
        unity when ‘phi’ for a particular grid point is equal to
        ‘phi0’. This occurs when you are located at the
        ‘standard_latitude’ for the case of a "secant" projection.
        Note that the grid spacing for a particular location in
        the domain is equal to ‘grid_spacing_m’/‘sigma’.


        Example 1: grid_spacing_m = 10000.
                   standard_latitude  =   +90.
```

```
            standard_latitude2 =   +90.
            grid_cen_lat =  +40.

            grid_spacing at projection (north) pole = 10km
            grid_spacing at domain center (+40)     ~  8km


Example 2:  grid_spacing_m = 10000.
            standard_latitude  =   +40.
            standard_latitude2 =   +90.
            grid_cen_lat   =   +40.

            grid_spacing at projection (north) pole ~ 12km
            grid_spacing at domain center (+40)     = 10km

Example 3:  grid_spacing_m = 10000.
            standard_latitude  =   -90.
            standard_latitude2 =   -90.
            grid_cen_lat   =   +40.

            grid_spacing at projection (south) pole = 10km
            grid_spacing at domain center (-40)     ~  8km

Example 4:  grid_spacing_m = 10000.
            standard_latitude  =   -40.
            standard_latitude2 =   -90.
            grid_cen_lat       =   -40.

            grid_spacing at projection (south) pole ~ 12km
            grid_spacing at domain center (-40)     = 10km


Note that the ‘Dx’ and ‘Dy’ values that appear in the
‘static.nest7grid’ should equal the value of
‘grid_spacing_m’.

Lambert:
    This is the latitude at which the grid spacing is exactly
    the nominal value (e.g. 10km). The projection cone will
    intersect the Earth’s surface at this latitude.
```

```
        Mercator:
            This is the latitude at which the grid spacing is exactly
            the nominal value (e.g. 10km). Equivalently this is the
            latitude at which the projection cylinder intersects the
            Earth.

6) standard_latitude2:

        Polar Stereographic:
            This must be set to +90. or -90. and defines the pole
            latitude of the polar stereographic projection
            (Earth's North or South Pole).

        Lambert:
            For a tangent lambert (e.g. CONUS), set this equal to
            the 'standard_latitude' parameter. For a secant
            (two-latitude) lambert, set this to the second true
            latitude where the projection cone intersects the surface.

        Mercator:
            N/A
```

When you run ./etc/localize_domain.pl, the netCDF static file 'static.nest7grid' will be automatically generated by process 'gridgen_model.exe'. This contains grids of latitude, longitude, elevation, and land (vs. water) fraction.

The following output message, "topo_30s file /U50N119W does not exist", does not necessarily mean there is a problem. It may signify that your domain runs outside the available 30" data, and should still be covered by the 10' worldwide data, if you are using the 'topo_30s' dataset. Other WARNINGs or ERRORs may be more significant.

3.5.2.1 Map Projection Functionality/Limitations

LAPS runs with the polar stereographic, lambert, and mercator projections. Please let us know if you encounter any problems.

The polar stereographic projection has a pole that may be set to either earth's north or south geographic poles.

Setting the pole to an arbitrary lat/lon (local stereographic) is a possible future enhancement. A test local stereographic domain gave an error of 2km in the grid points; the test code works in cases where the projection pole coincides with the center of the domain. Further improvement of this may include more fully converting library subroutines 'GETOPS' and (possibly) 'PSTOGE' to double precision.

The projection rotation routine 'projrot_laps' also has some approximations when local stereographic is used. These need to be checked for their validity and refined if needed. Cases of interest include a projection pole point at the domain center, as well as offset from the center.

The local stereographic projection also ignores 'standard_latitude' from the namelist so this is internally assumed to be +90. This means that the grid spacing is valid at the projection pole location, regardless of both where on the earth the pole is and the pole's latitude.

The map projection calculations are performed with a spherical earth assumption.

3.5.3 Domain Resolution

The default value of the 'grid_spacing_m' parameter is 10000m. This is one of the parameters used in constructing the static file (as mentioned above). To date, we have run LAPS with resolutions ranging from 1000m to 48000m.

3.5.4 Terrain Smoothing/Filtering

Edit the file 'data/static/nest7grid.parms'...

```
1) silavwt_parm:

   Default value of 0. This parameter allows the potential use of
   silhouette terrain which is the maximum elevation in the local area.
   Useful range is anywhere between 0-1. A value of zero uses the
   average terrain instead of the maximum. Note that a value of 1
   may reduce the apparent effect of filtering with 'toptwvl_parm'.

2) toptwvl_parm:

   For example, a value of 4 represents 4 delta-x filtering of the
   terrain. You can change this to alter the smoothness of the
   terrain. Higher numbers mean smoother terrain.
```

3.6 Changing the Vertical Domain

PRESSURE OF THE LEVELS (and vertical resolution):
To do this, perform the following between untarring the tar file and localizing LAPS

- Copy 'data/static/pressures.nl' to your $TEMPLATE directory, then edit it with to have the new set of levels.
- Update the list of pressures that go in sequence from higher to lower pressures (bottom to top)

Note that the default vertical grid uses constant pressure coordinates and that the vertical pressure interval can vary between levels. For example one might want to use higher density in the boundary layer (100Pa interval) and make it coarser higher up (250Pa interval).
Of course the top pressure should be greater than zero mb. The bottom level should extend below the terrain and below the observations. The pressure values must be in multiples of 100 pascals, corresponding to an integer number of millibars.
NUMBER OF LEVELS:

1. The default value of 'nk_laps' is set to 21 levels in 'data/static/nest7grid.parms' and will automatically be reset during the localization (based on the contents of 'pressures.nl').

2. Note that compatibility with model background data will depend of the vertical extent of that data source.

Note: If you are feeding LAPS output into an AWIPS workstation, then additional workstation related changes may be needed.

3.6.1 Sigma Height Grid

UNDER DEVELOPMENT - mainly for STMAS-3D
Similar to 2.8 except that one changes the 'vertical_grid' parameter in 'nest7grid.parms'. Also the 'heights.nl' namelist is used instead of 'pressures.nl'. Note the heights in this namelist are scaled sigma values where the namelist (idealized) height = sigma * (height_top - height_bottom) Presently the height_top and height_bottom values are hard wired to 20000. and 0. meters, respectively.

3.6.2 Sigma Pressure Grid

UNDER DEVELOPMENT - mainly for STMAS-3D
Similar to 2.8 except that one changes the 'vertical_grid' parameter in 'nest7grid.parms'. Also the 'sigmas.nl' namelist is used instead of 'pressures.nl'.

3.7 Changing the Cycle Time

The default cycle time is 60 minutes. To change this, do as follows...
edit runtime parameter file 'data/static/nest7grid.parms' to change the value of 'laps_cycle_time'.

3.8 LQ3 (Humidity Analysis) Changes

Recent changes as of February 26, 2006 NAMELIST

The namelist file ./laps/static/moisture_switch.nl controls the data assimilation within the moisture analysis. This file is self-documented, refer to it for details. This file has not changed in this latest update however, one of it's controlling aspects is GVAP or GOES vapor (total precipitable water, product data) and the application of this data has changed since a major implementation change March 2005.

OPTRAN

The NESDIS Community Radiative Transmittance Model (CRTM) and forward radiance model called OPTRAN is incorporated into the current release of LAPS. Details of OPTRAN are available from: Tom Kleespies NOAA/NESDIS Thomas.J.Kleespies@noaa.gov

Also OPTRAN can be used by any U.S. Government or U.S. Military entity without problem. ALL other users need to contact NESDIS (Tom Kleespies) and receive authorization to use this software. Generally a simple acknowledgement to give full credit to the program author is all that is required. GSD assumes no obligation or responsibility in integrating this software as part of LAPS. To disable the use of OPTRAN in LAPS, simply assign the GOES option in the moisture_switch.nl namelist file to zero.

The version of OPTRAN in LAPS is configured to work with GOES-8 and -10 sounder or imager at this time. Note also that GOES imager channel 5 (water vapor split window) is currently not available on GOES 11, 12 and future satellites since it was replaced with a different band. Therefore, the GOES imager data should not be used in the moisture algorithm for any GOES satellite 11 and beyond. There are simply not enough moisture channels available to offer any useful information about moisture depth due to this change. Furthermore sounder radiances for GOES-10 are deemed about 98% reliable, they are regarded to be 100% reliable for GOES-8. NaN values have been observed being generated from the GOES-10 sounder coefficients that currently accompany this software. At this time there are only basic provisions to handle the NaN state conditions. They have not been observed to crash the moisture analysis and seem to be handled gracefully to date. Any observation otherwise needs to be communicated to:

Dan Birkenheuer NOAA/GSD Daniel.L.Birkenheuer@noaa.gov

To model the atmosphere with OPTRAN, an atmosphere is formulated that extends to 0.1 hPa. This is a composite of the normal LAPS analyzed vertical domain (nominally extending to 100 hPa), spliced together with a climatological atmosphere of 20 levels that extends to 0.1 hPa. The joining of the two vertical coordinate systems is computed automatically and is continuous. This will automatically take place even if the nominal LAPS levels are extended beyond 100 hPa. In this upper region, temperature, and mixing ratio are functions of latitude and Julian day. Ozone is based on the U.S. Standard Atmosphere.

ADDENDUM: routine RAOB_STEP.F

It should be noted that some users have had to modify the parameter that defines dimensions in routine raob_step.f due to the fact that this can overflow array limits on some machines. The current parameter snd_tot is set to 1000. The primary reason for this is to accommodate satellite soundings of which there can be many in even a small area. This parameter ties in to the dimensions of the weight matrix (ii,jj,snd_tot). If a large horizontal domain is defined, and you don't have a lot of RAOB data and are not using satellite processed soundings, you may have better success at compiling this routine by reducing the value of snd_tot to a smaller value.

GVAP

GVAP data are GOES sounder total precipitable water data acquired from the sounding retrieval process. These data were added to LAPS under a grant from NOAA NESDIS.

The analysis for GVAP data has recently changed from the prior application. Up until the March 2005 release, GVAP data were used as a direct total moisture data source in that the integrated state variable in the moisture routine (q) was compared to GVAP totals and part of the minimization procedure was to improve this match through variational techniques.

It was learned during the IHOP 2002 experiment that the GVAP data were badly biased, especially at asynoptic times. (see http://laps.noaa.gov/cgi/birk.pubs.cgi for all publications, and specifically broken link for the article about IHOP, or it can be located in the literature at: Birkenheuer, D., and S. Gutman, 2005: A comparison of the GOES moisture-derived product and GPS-IPW during IHOP. J. Atmos. Oceanic Tech. 22, 1840-1847.)

As a result, the algorithm was modified to use GVAP gradients and to compute gradients in the solution field and match these gradients to those from GVAP. The advantage to using gradients in this procedure was that it eliminates bias and improves data structure. There is not a Tech Memo that has been published and is also available on line that describes this new technique. (refer to: http://laps.fsl.noaa.gov/birk/papers/tech_memos/GSD_Tech_Memo_32.pdf or a copy can be gotten directly from GSD)

3.9 Other Runtime Parameters

It is worthwhile to check the 'nest7grid.parms' and other namelist files in 'data/static' to make sure all the runtime parameters are correct. Some parameters worth noting are:

```
nest7grid.parms
---------------

c8_project_common - Depends on which "realization" of LAPS you are running.
                    Allowed values are listed within 'nest7grid.parms'.



cloud.nl
--------

l_use_vis -         Boolean set to indicate whether we are confident in the
                    calibration of the visible satellite data and albedo
                    fields for use in the cloud analysis. This is normally
                    set to .true. at GSD and .false. for WFO and other
                    ports unless we are confident in the vis data
                    normalization.
```

3.10 Detecting and Reporting Installation Errors

To determine how well LAPS was installed, verify that all (31 at last check) executables were built OK ('bin' directory) with no errors in the output of 'make'.
Similarly, check the output of the localization script.
If you have any problems during the configure, install and localization process, there are several things to check. For certain platforms, you can compare your build output with ours by clicking on "Results of Latest LAPS Builds" on the LAPS Software page. Also double check that you've followed all the installation steps in this section of the README. There is also a FAQ available at http://laps.noaa.gov/birk/LAPS_FACTS.htm Finally, check the release notes at the http://laps.noaa.gov/software/release_notes.html URL.
If you don't find the answer in these documents, send mail to oplapb.gsd@noaa.gov Include in your mail:

```
LAPS version number (hopefully you're using the latest version?)
The type of system (often, uname -a)
The system limits (ulimit -a)
The applicable compiler versions (often a -v or -V option to the compiler)
The entire output of configure
The entire output of make (standard output + error output)
The entire output of localize_domain.pl
                      (found in $LAPS_DATA_ROOT/log/localize_domain.log*)
```

3.10.1 Runtime Monitoring

To see how well LAPS is running, check if output files are being placed in the various 'lapsprd' subdirectories. A graphical product monitor that can help with this is available in 'etc/laps_monitor.pl'. This script may need some simple editing to suit your needs (e.g. to specify the $LAPS_DATA_ROOT[s]). The monitor script writes HTML output to 'stdout'. This HTML output, if routed to a file or hooked up to a Web server, can be viewed with a browser. You can click on http://laps.noaa.gov/monitors/Laps_Monitor.cgi to see an example of the monitor output. Green means optimum product continuity, red means the product is failing to generate, yellow means it is generating OK now but has failed in the past. The numbers in the columns indicate the number of files in each directory, as well as the youngest and oldest file ages in hours. The data flow is generally from top to bottom on the product list, starting with analyses and ending with forecast model (fua/fsf) output being listed at the very bottom.
To check what model background and observational data were used in the analyses as well as some QC and error (verification) statistics, you can view the log file summaries in the files '$LAPS_DATA_ROOT/log/*.wgi.yydddhhmm'. Generally each named 'wgi' file corresponds to the name of an analysis process, except that 'sfc.wgi.*' is generated by one of several executables than can provide the LAPS surface analysis depending on the runtime configuration.

For more details, check the various log files in the '$LAPS_DATA_ROOT/log' directory for occurrences of the string 'error' and 'warning'. The errors are generally more significant. If any core dumps occur they can usually be flagged by searching for the 'sh:' string in 'sched.log.*'.
If you are reporting runtime errors it can be useful to tar up your entire $LAPS_DATA_ROOT and make it available on your web or FTP server as a compressed "tgz" file. If the data set is very large you might consider mailing us a CD or DVD. Alternatively if you have the untarred $LAPS_DATA_ROOT files on your web server we can browse through the directories for the log and product files as needed to help diagnose the run.
If the $LAPS_DATA_ROOT is large it can be pared down as there is a script called 'etc/tarlapstime.sh' that works by just tarring up the current hour's worth of files. If you need to narrow this down further just the inputs to the particular analysis would be needed as shown in section 4 of the README. Also, things like the 'cdl', 'time' and 'static' subdirectories should be included.

Chapter 4

Description of LAPS Processes

The following section contains information on which LAPS processes generate which LAPS output products. Static data (like lat and lon grids) are included in section 4.1. These are the processes contained within the LAPS tar file and built with the localization script.
"Inter data" is an ascii file containing non-gridded data (intermediate data files). Examples of this are surface obs, profiler obs, etc.
This list contains all outputs generated by LAPS processes.
The products listed under each process are the outputs produced by that process. Inputs are listed here for some analyses. If the cron including 'sched.pl' (see section 3.2) is run according to the flow therein, the necessary inputs will be available.

4.1 Localization Processes

4.1.1 Gridgen_model (static.nest7grid generation)

Package: gridgen_model.exe - Writes static file, run by localization script.
Contact: Steve Albers - Steve.Albers@noaa.gov
Inputs: Geography databases (topography, land fraction, landuse, soiltype top/bot) greenness fraction, mean annual soil temperature, and albedo. Files are typically in 10, 30, or 180 deg tiles. See section 2.2.5 for details on the geography data. static/nest7grid.parms

```
Outputs:
          static/static.nest7grid   netCDF grid   geography data mapped to LAPS
                                                grid

                                        'LAT'  latitude in degrees
                                        'LON'  longitude in degrees
                                        'AVG'  mean elevation MSL
                                        'LDF'  land fraction (1.0=land; 0.0=water)
                                        'LND'  land-water mask (1=land; 0=water)
                                        'USE'  usgs 30s (24 vegetation categories)
```

```
     landuse data (currently dominant category
     for each grid point).

 1:   Urban and Built-Up Land
 2:   Dryland Cropland and Pasture
 3:   Irrigated Cropland and Pasture
 4:   Mixed Dryland/Irrigated Cropland and Pasture
 5:   Cropland/Grassland Mosaic
 6:   Cropland/Woodland Mosaic
 7:   Grassland
 8:   Shrubland
 9:   Mixed Shrubland/Grassland
10:   Savanna
11:   Deciduous Broadleaf Forest
12:   Deciduous Needleleaf Forest
13:   Evergreen Broadleaf Forest
14:   Evergreen Needleleaf Forest
15:   Mixed Forest
16:   Water Bodies
17:   Herbaceous Wetland
18:   Wooded Wetland
19:   Barren or Sparsely Vegetated
20:   Herbaceous Tundra
21:   Wooded Tundra
22:   Mixed Tundra
23:   Bare Ground Tundra
24:   Snow or Ice

'U01-U24' Fractional distribution of landuse category
          (not active).
'STL'     Soil type - top layer (0-30cm)
'SBL'     Soil type - bottom layer (30-90cm)
          (currently dominant category for each grid pt)

FAO/WMO 16-category soil texture:
          1          SAND
          2          LOAMY SAND
          3          SANDY LOAM
          4          SILT LOAM
          5          SILT
          6          LOAM
```

```
                                    7          SANDY CLAY LOAM
                                    8          SILTY CLAY LOAM
                                    9          CLAY LOAM
                                   10          SANDY CLAY
                                   11          SILTY CLAY
                                   12          CLAY
                                   13          ORGANIC MATERIALS
                                   14          WATER
                                   15          BEDROCK
                                   16          OTHER (land-ice)

                                 'T01-T16' Fractional distribution of top layer
                                           soil texture class (not active)
                                 'B01-B16' Fractional distribution of bottom layer
                                           soil texture class (not active)
                                 'TMP'     Mean annual soil temperature
                                 'G01-G12' Monthly (center of month) greenness fraction
                                 'A01-A12' Monthly (center of month) albedo climatology
                                 'ALB'     Not used

static/latlon.dat         Binary grid   latitude
                                        longitude

static/topo.dat           Binary grid   mean elevation

static/corners.dat        ASCII         lat/lon of 4 corner points

static/latlon2d.dat       ASCII         latitude/longitude
```

Source directory: laps/src/grid
Sample Output: Should be available in the test data case. The grids start with gridpoint (1,1) in southwest corner of the domain and end with gridpoint (ni,nj) in the northeast corner. The bottom (southernmost) row of the domain is written first (I increases with consecutive grid points, then J increases). I increases as you're moving east on the grid, J increases as you're moving north.

4.1.2 Surface Lookup Tables (gensfclut.exe)

Package: gensfclut.exe - Writes surface lookup tables, run by localization script. (contact: John McGinley / Steve Albers)
Source directory: laps/src/sfc/table

```
Output:
```

```
        static/drag_coef.dat      Binary grid   Drag Coefficients
```

In 'gensfclut.exe', the friction parameter has been configured by automatically producing a scaling factor based on the range of elevations across the domain. This factor can be changed in the 'drag_coef' section of 'build_sfc_static.f', if so desired.

4.1.3 Satellite Lookup Tables (genlvdlut.exe)

Package: genlvdlut.exe - Writes satellite lookup tables, run by localization script. (contact: John Smart)
Source directory: laps/src/ingest/satellite/lvd/table

```
Output:
        static/lvd/*.lut                     Satellite Lookup tables
```

Additional information on the lookup tables can be found in the file 'laps/src/ingest/satellite/README'.

4.2 Ingest Processes

As mentioned above, a flow chart for the ingest processes ican be viewed in 3.1 or may be found at http://laps.noaa.gov/doc/slide1_v3.gif.

4.2.1 LGA Model

Package: lga.exe - ingest background model data (contact: Steve Albers - Steve.Albers@noaa.gov).
LGA LAPS analysis grids from RUC or other analysis/forecast grids.
Inputs: Raw model data on the model's native grid. The acceptable models and formats for the background model are listed in 'data/static/background.nl'. We have recently added support for the background models to include GRIB-formatted files. See source directory: $LAPS_SRC_ROOT/src/lib/degrib/README_LIBS file for detailed information.
For some models you might want to do a separate conversion of GRIB to netCDF prior to running LGA. One software option for this is available from the GSD/ITS group as described in section 4 of this web document at the following URL:

http://www-its.fsl.noaa.gov/dsg/FSL-WhitePaper.html

Tropical cyclone bogusing information is also an input in the form of the 'tcbogus.nl' namelists. These are generated independently of LAPS as "raw" data, yet are placed in the '$LAPS_DATA_ROOT/lapsprd/
subdirectory. The filename convention should be 'yydddhhmm_tcbogus.nl'. To see the format check the sample file located in 'data/static/tcbogus.nl'.
Outputs: (Feeds various analyses)

```
lga grid background model 3-D data analysis/forecast
lgb grid background model Sfc data analysis/forecast
```

Source directory: The source code for this is in 'src/background'.
Library directory: Associated library modules are in 'src/lib/bgdata'.
Parameter namelist file: 'static/background.nl'
Sample Input/Output: May be available in the test data case.
This software currently supports nearly 10 different models. If additional models are required, then software mods may be needed, potentially a new source file added to 'src/lib/bgdata/read*.f'. A key variable that relates to which model you're using is 'bgmodel'.
Note that time interpolation is used if the required LAPS analysis time(s) are between the valid forecast times for two of the set of input files. In this context output files are produced for the LAPS analysis time as well as +/- one LAPS analysis cycle time. Input data for LGA should thus be available over an appropriate time span.

4.2.2 Surface Data Ingest

4.2.2.1 obs_driver.x

LSO process - obs_driver.x - Ingest surface data (author: Pete Stamus/Steve Albers)

```
Input:
METAR/SYNOP data
        Buoy/ship (maritime) data
MADIS (or LDAD) mesonet/urbanet data
        GPS (sfc obs - without precipitable water)
        Profiler surface data (via LDAD)


Output:
         LSO ascii LAPS surface obs intermediate data file: METAR, Mesonet,
                        and Buoy/ship obs. The format of this file may be
                        determined from looking at library access routines such
                        as 'read_surface_data', the obs_driver code, or by
                        looking at sample LSO files in the test data case.


Sample Input/Output: May be available in the test data case.
```

Source directory: $LAPS_SRC_ROOT/src/ingest/sao (contains a README file)
Parameter file (specifies input data paths and formats): 'static/obs_driver.nl'
The LSO file is fairly self explanatory. The easiest way to see what goes where is to look at the routine 'read_surface_data' in the file 'src/lib/read_surface_obs.f', and the corresponding format statements in the file 'src/include/lso_formats.inc'.
The routines are pretty well commented, and should be enough to tell you what you need to know if you want to make a decoder that outputs an LSO-type formatted file directly. This direct route would allow you to bypass the step of producing "raw" netCDF surface observation data.

Here are a few recommended settings for the observation type variables (reportType and autoStationType) if you are constructing your own LSO file:

```
raw data    reportType    autoStationType

--------    ----------    ---------------
 metar       METAR         UNK (unless an automated A01 or A02 station)
 synop       SYNOP         UNK (unless an automated A01 or A02 station)
 buoy        MARTIM        FIX
 ship        MARTIM        MVG
```

The expected accuracies are based on "offical" NWS numbers where possible. For LDAD observations, they're just a best guess, since no one really knows how good the obs are. These expected accuracies will be used in the quality control routines sometime in the future. The lat/lons are in decimal degrees.
Gross "climatological" QC error checks are applied to several variables including temperature, wind, and pressure. MADIS QC flags are checked as can be controlled via namelist.

4.2.2.2 sfc_qc.exe

Process: (sfc_qc.exe) LAPS Surface Ingest Quality Control

```
LSO QC process - sfc_qc.exe - QC the ingest surface data
        (author: John McGinley / Pete Stamus)


Input:
 lso    ascii


Outputs:
         lso_qc ascii QC'd LAPS surface obs file: METAR, Mesonet and
                      Buoy/ship obs.


         lsq    netCDF  QC'd LAPS surface obs in NIMBUS netCDF format


         lsq/monster*.dat' are binary files that contains internalinformation
         about the data history
```

`Sample Input/Output: May be available in the test data case.`

Source directory: 'laps/src/ingest/sfc_qc'
The 'lso_qc' file is fairly self explanatory. . . the comments in the 'write_surface_obs' routine detail everything. The expected accuracies are based on "offical" NWS numbers where possible. For LDAD observations, they're just a best guess, since no one really knows how good the obs are. These expected accuracies will be used in the quality control routines sometime in the future. The lat/lons are in decimal degrees.

This new QC package compares the observations temporally and fills in predicted values for an observation when it is only intermittently available. This helps compensate for temporal changes in data density. A future enhancement may be to replicate cloud layer observations for a few hours. This would be especially helpful when running an hourly LAPS while SYNOP stations report clouds only every 3 hours.
More information is in the NWP conference paper on Kalman Filtering (McGinley 2001) at http://laps.noaa.gov/cgi/LAPB.pubs_01.cgi

4.2.2.3 How to Blacklist Stations

author: Steve Albers/Pete Stamus
As part of the 'obs_driver' code, a Blacklisting function has been added. This allows users to tell LAPS to skip stations with known bad variables (one or several), or to skip a station completely. As of this writing, the user will have to edit a "Blacklist.dat" file…in the future we hope to include this function in the LAPS GUI.
An example file, called "Blacklist.example" has been included in the same directory as this README file. It shows the format that *must* be followed for the Blacklist to work properly. An error in the format will either allow the bad station(s) through, or crash the program completely. Let's decode the "Blacklist.example" file:
The first line is the number of obs to blacklist…in this case, 5. Each station goes on a new line. The number of variables to blacklist for that station is next, then the codes for the variable(s) follow. For the first station (KFCS), we are blacklisting the 3 pressure variables. To blacklist the entire station (KDTW) use 1 for the number of variables, and "ALL" as the variable. All the stations from a particular provider can be blacklisted by adding 100 to the number of variables (third example). The last two examples show 1 and 2 individual variables, respectively.
These are the valid codes for variables to blacklist:

```
"ALL" - Set all variables at this station to bad
"TMP" - Set temperature bad
"DEW" - Set dew point temperature bad
"HUM" - Set relative humidity bad
"WND" - Set wind bad (this does both speed and direction, and gusts)
"ALT" - Set altimeter bad
"STP" - Set station pressure bad
"MSL" - Set MSL pressure bad
"VIS" - Set visibility bad
"CLD" - Set clouds to bad (this does all cloud layers reported)
"PCP" - Set precipitation amount to bad (all reported, 1-12 hrs)
"SNW" - Set snow cover to bad
"SOL" - Set solar to bad (if reported)
"SWT" - Set soil/water temperature to bad (if reported)
"SWM" - Set soil moisture to bad (if reported)
```

You might keep in mind that some variables act as a group. For example both "HUM" and "DEW" variables feed into the LAPS dewpoint analysis so consideration should be given as to whether to blacklist one or both of these variables. "ALT", "STP" and "MSL" are a similar group of pressure variables. An incorrect variable code generates a warning message, and the code should hopefully continue without acting on the station in question.

Note that when a station is blacklisted, its name, latitude, longitude, elevation, and time, will still be stored in the LSO file. However, the selected variables (up to "ALL" of them) will be set to the 'badflag' value and skipped in the analyses.

To actually get this stuff working, edit the file called "Blacklist.dat" in the 'data/static' (or template) directory. The "Blacklist.dat" being used at GSD is supplied in this directory as a default, and this provides additional examples. Format the file *exactly* as the 'Blacklist.example' file (using your station information, of course). Save the file, and the next time 'obs_driver' runs, it will use the blacklist information. This will be noted in the 'obs_driver.log' file.

You may eliminate element specific or ALL data from a particular provider by replacing the leading 0 with a 1 in the second column. See the WXforYou example in the Blacklist.example file. To ensure the elimination of the data by provider, care must be taken to make certain the correct provider is listed in the Blacklist.dat file.

Primarily, offending datasets are from stations received through LDAD. To find the provider for a given station you can look in the "log/sfc.wgi.*" files or in the input LDAD netCDF files. For AWIPS users a list of these stations are kept in ldad/data on your AWIPS system. Each .txt file in ldad/data will have a .desc file associated with it which describes the data being ingested etc., by that provider. Look in a particular .desc file of interest. Go to the 3rd word of the 1st line which is not commented out (e.g. aprswxnet : -9999.00 — APRSWXNET). For this example, the APRSWXNET (3rd word) is the provider name and should be the entry used if utilizing the elimination by provider feature of the Blacklist.

4.2.3 Polar Radar Data (e.g. WSR 88D Level II, Level III)

Process: remap_polar_netcdf.exe

Author: Steve Albers (Steve.Albers@noaa.gov)

Every volume scan Initiation: Completion of volume scan

Inputs: Wideband Radar Data (reflectivity and velocity in polar coordinates, in netCDF format) These have one tilt per file and at least 4 tilts per volume scan (all with the same volume timestamp in the filenames). This data can be obtained from a WSR 88-D Level-II data feed or the equivalent. A description of how we obtain these Polar netCDF files for Level-II is at http://laps.noaa.gov/albers/remapper_raw.html .

Note that narrowband reflectivity data (e.g. WSR 88D Level-III RPG) can also be used as long as it is converted to the required polar coordinate, netCDF format. This is in fact being done for the AWIPS implementation of LAPS for a low-level tilt from a single radar, via the 'etc/LapsRadar.pl' script running in the AWIPS environment. The comment section at the top of this script explains how this 4 bit processing of reflectivity data works. 'etc/LapsRadar.pl' runs two executables. The first executable 'tfrNarrowband2netCDF' from AWIPS, writes out the polar netCDF files in the directory '$LAPS_DATA_ROOT/lapsprd/rdr/???/raw' where ??? is the radar number. The second executable 'remap_polar_netcdf.exe' is run as part of LAPS. We haven't been using Level-III velocity data since it is of limited 4-bit resolution and we're running only with the lowest tilt at present.

For both Level-II and Level-III the polar netCDF files are named according to 'yydddhhmm_elevxx' where 'xx' is the tilt number. Sample polar netCDF files including a CDL may be found at: http://laps.noaa.gov/software/radar/wideband/

Outputs (LAPS intermediate files - depending on input parameters):

```
v01          grids       3-D Radar reflectivity, velocity, and Nyquist vel
v02            "                  "
rdr/???/vrc    "         2-D Radar reflectivity  (??? = radar number)
vrc            "                  "
etc. (for each radar)
```

The outputs from this process, on the Cartesian LAPS grid, are used by the LAPS wind analysis, and also potentially by cloud and precip accumulation analyses. One output file is written per volume scan.

When running the remapper, files such as v01, v02, vrc, etc. are produced depending on which radar is being used and on the input parameters. A further description of how the remapper software functions may be found on the World Wide Web at http://laps.noaa.gov/albers/remapper_laps.html. Also recall the flow chart showing the inputs and outputs for 'remap_polar_netcdf.exe' at http://laps.noaa.gov/albers/laps/radar/laps_radar_ingest.html.

Source directory: The source code for this is in 'src/ingest/radar/remap'.

Compile time parameters: 'src/include/remap_dims.inc'

Runtime parameter namelist file: 'static/remap.nl'

Sample Input/Output: May be available in the test data case.

4.2.4 WSI/NOWRAD Radar Preprocessing (VRC)

Process: VRC (vrc_driver.x)

Author Steve Albers (Steve.Albers@noaa.gov)

Parameter namelist file: 'static/vrc.nl'

```
Inputs:
        Raw WSI NOWRAD radar reflectivity data


Outputs (Intermediate data file):
        vrc     grid                         2-D reflectivity
```

The WSI data are decoded externally to LAPS and written as netCDF files in NIMBUS format. The vrc_driver.x process reads these netCDF files. WSI sends out many types of radar data. We use the files that are labeled "_hd" (15 min freq). They also send out an "_hf" (5 min freq)

file. We use hd because WSI hand edits these for ground clutter. The hf files are not edited. The hd and hf files are composites of "low-level" elevation scans from the 88D's around the country. The vrc_driver.x also maps from conus to laps domain for the wfo data set. The map transformation software is found in lib/gridconv, lib/nav, and lib/radar/wsi_ingest. The switch to use wsi versus wfo in variable c_raddat_type in the 'nest7grid.parms' namelist. Pathway to data is variable path_to_wsi_2d_radar in 'vrc.nl'.
The output reflectivity is used by the cloud and precip accumulation analyses.

4.2.5 Radar Mosaic

Process: (mosaic_radar.exe)
Author Steve Albers (Steve.Albers@noaa.gov)

```
Inputs:
        v01, v02, etc.                        3-D reflectivity
        rdr/001/vrc, rdr/002/vrc, etc.        2-D reflectivity

Outputs (Radar Mosaic - intermediate data file):
        vrz     grid                          3-D reflectivity
        vrc     grid                          2-D reflectivity
```

This program runs once per LAPS cycle in the 'sched.pl'. The default is to write just one mosaic file for the cycle valid at 'systime'. A namelist option allows this program to produce multiple mosaic outputs within a given LAPS cycle. The multiple mosaics are all run at the same wall clock time, while the valid mosaic times are spaced throughout the previous LAPS cycle.
The nearest radar with valid data is the one chosen to contribute at each grid-point.
The output reflectivity mosaic is used by the cloud and precip accumulation analyses. Further QC is done within these analyses.
Parameter namelist file: 'static/radar_mosaic.nl'

4.2.6 PROFILER/VAD/SODAR (PRO) Ingest

Process: PRO (ingest_pro.exe) LAPS Wind Profile Ingest
Author: Steve Albers (Steve.Albers@noaa.gov)

```
Inputs: (located in separate netCDF directories)
NPN 404-MHz profiler wind data in netCDF format
            NetCDF CDLs from both GSD-NIMBUS and AWIPS/MADIS are accepted.

Boundary layer 915-MHz profiler wind data in netCDF format
            (GSD-NIMBUS/DD, AWIPS-LDAD & MADIS-MAP CDLs).

        50-MHz profiler in netCDF (AWIPS-LDAD & MADIS-MAP CDLs).
```

```
Doppler Radar VADs in netCDF (GSD-NIMBUS CDL) format

        SODAR data in netCDF format (AWIPS-LDAD & MADIS-MAP CDLs).

Output: (feeds wind)
pro   inter data   wind profile direction and speed (ASCII-true North)
```

Source directory: laps/src/ingest/profiler
Parameter namelist files: static/nest7grid.parms, static/vad.nl
Sample Input/Output: Should be available in the test data case.
For the 'pro' output, each profile starts with an ASCII header and the formatted entries are defined in sequence

```
1)  WMO ID or other ID number. The use of this is optional and zero can
    be used if you don't know the number.
2)  Total number of levels for which data is provided. This can include
    the surface data as the first level.
3)  Latitude (degrees)
4)  Longitude (degrees)
5)  Station Elevation (meters MSL)
6)  Station Name
7)  Time of observation (UTC). This is the middle of the observation period
    if time averaging is used.
8)  Data type. Can be either "PROFILER" or "VAD"
```

After the header, the data entered for each level is as follows

```
1)  Elevation (meters MSL)
2)  Wind Direction (degrees)
3)  Wind Speed (meters/second)
4)  Estimated Root Mean Square (RMS) error of measurement
```

4.2.7 RASSs (LRS) Ingest

Process: (ingest_lrs.exe) LAPS local data RASS ingest
Author: Steve Albers (Steve.Albers@noaa.gov)

```
Inputs:
NPN RASS temperature data in NIMBUS netCDF format
Boundary layer RASS data in netCDF formats
            (NIMBUS, RSA-LDAD, MADIS-MAP)

Outputs: (feeds LSX and temp.exe)
lrs inter data RASS Virtual Temperatures (ASCII)
```

Source directory: laps/src/ingest/rass
Sample Input/Output: Should be available in the test data case.

4.2.8 PIREPS/ACARS Ingest

Process: (ingest_aircraft.exe) LAPS Pireps / ACARS
Author: Steve Albers (Steve.Albers@noaa.gov)

```
Inputs:
Aircraft voice pireps (cloud layer reports)
            NetCDF files using GSD-NIMBUS or WFO/AWIPS CDLs

        ACARS data
            NetCDF files using GSD-NIMBUS CDLs
                Uses pressure altitude
                Hourly netCDF filename convention is ‘yydddhh00q.cdf’

            NetCDF files using MADIS CDLs
                Same format as GSD-NIMBUS
                Software can automatically switch to the MADIS filename
                convention of ‘yyyymmdd_hhmm’ if needed.

            NetCDF files using WFO/AWIPS CDLs
                Uses pressure altitude

            AFWA ASCII format also allowed for ACARS

        WISDOM Balloon Data
            NetCDF files using MADIS CDLs

Outputs: (Intermediate output written to the ‘pin’ file. Feeds cloud.exe,
          wind.exe, lq3driver.x)

pin inter data voice pireps/clouds
                                ACARS/(wind, temp, mixing ratio - using
                                                             pressure altitude)
```

Source directory: The source code for this is in ‘src/ingest/acars’.
Parameter namelist file (for data paths): ‘static/nest7grid.parms’
Sample Input/Output: Should be available in the test data case

4.2.9 Sounding (RAOB/Dropsonde/Sat/Radiometer) (SND) Ingest

Process: (ingest_sounding.exe) LAPS Soundings
Author: Steve Albers (Steve.Albers@noaa.gov)

```
Inputs:
RAOB in various formats:
                                 (GSD-NIMBUS CDL - netCDF)
                                 (WFO/AWIPS CDL - netCDF)
                                 (AFWA and CWB ASCII formats also allowed)

        Satellite Soundings (GOES binary, MADIS POES, or AFWA format)

        Dropsonde (AVAPS, CWB, or SND format)

        Radiometer (GSD-MADIS CDL - netCDF)

        Met Tower (LDAD netCDF)

Outputs: (Feeds temp.exe, humid.exe, wind.exe)
snd inter data (ASCII)   sounding temp, dewpoint, wind (true N)
```

Source directory: laps/src/ingest/raob (contains a README file)
Parameter namelist file (for data paths): 'static/snd.nl'
Sample Input/Output: May be available in the test data case. If not, the README in the source directory contains a description of the 'snd' file.
Note: Sounding data is used if the observations lie in the time window of +/- 'laps_cycle_time' centered on the analysis time. There are flags to toggle usage of the sounding (i.e. snd) data in 'wind.nl', 'temp.nl' and 'moisture_switch.nl'.

4.2.10 LVD (Satellite Image + Cloud Top Pressure)

LVD process - lvd_sat_ingest.exe - takes raw sat. data and puts it on LAPS grid. (author: John Smart, contact Kirk Holub - Kirk.L.Holub@noaa.gov)
Input: GOES or other satellite data

```
Output:
         LVD/'SATID'grid LAPS satellite data file
SATID (e.g. goes08 or goes09)


         CTP            grid    Cloud-top pressure information
```

Parameter namelist file: 'static/satellite_lvd.nl'
Source directory: laps/src/ingest/satellite/lvd (contains a README file)

4.2.11 Cloud Drift Wind (CDW) Ingest

Process: (ingest_cloud_drift.exe) LAPS Cloud Drift Winds
Author: Steve Albers (Steve.Albers@noaa.gov)

```
Inputs (supported cloud drift wind formats):
GOES cloud drift winds in NESDIS (ASCII) format
        MADIS netCDF format
        AFWA & CWB formats are also allowed.


Outputs: (Feeds wind.exe)
cdw intermediate data (ASCII)       Satellte cloud drift winds
```

Parameter namelist file: 'static/cloud_drift.nl'
Source directory: laps/src/ingest/satellite/cloud_drift
Note: Sounding data is used if the observations lie in the time window of +/- 'laps_cycle_time' centered on the analysis time.

4.3 Analysis Processes

A flow chart for the analysis processes is shown in figure 4.1 or may be found at this URL: http://laps.noaa.gov/doc/LAPS_flow_v_02.png Listed below is a summary of each analysis process in the order it is typically run by the 'sched.pl' script.

4.3.1 WIND

Process: wind.exe - WIND analysis and related fields
Author: Steve Albers (Steve.Albers@noaa.gov)

Generate a wind analysis using surface observations, profiler, cloud drift wind, and aircraft reports. VAD and SODAR can also be read in. Background model grids are used as a first guess and to do quality control on new observations. Time tendencies from the background model are applied to the aircraft/cloud-drift wind reports when they are taken before or after the nominal analysis time. The quality control rejects any observations deviating from the background by more than a threshold depending on observation type as in the following table.

```
        ACARS               10 m/s
        Cloud-Drift winds   10 m/s
        Profiler            22 m/s
        Doppler Radar       12 m/s
        Other               30 m/s
```

The wind analysis is done in three steps. The first step analyzes the non-radar data with the background wind field using a multiple iteration successive correction technique.

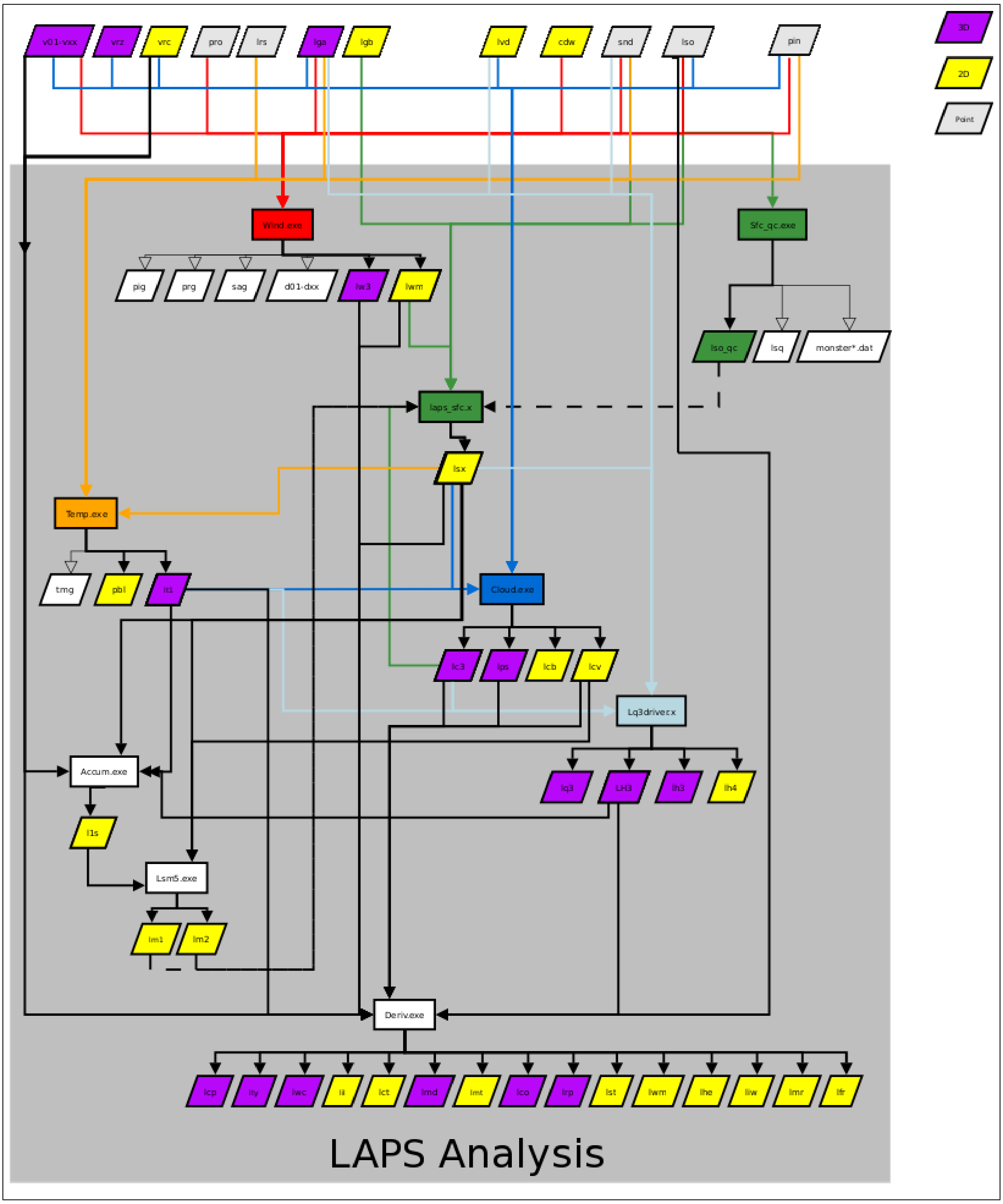

Figure 4.1: Flow chart for the analysis processes

For the second step, the first step results are used as the background. The data used includes non-radar data; any grid-points with multiple- Doppler radial velocities are also mixed in. Radial velocities are taken from the Doppler radars after dealiasing and other quality control steps are done. If two or more radars illuminate a given grid-point, a full wind-vector is constructed from a combination of the radial velocities and the preliminary non-radar analysis. This is done via a “successive insertion” process, beginning with the background (non-radar analysis), then followed with the radial velocity from each radar in sequence.

For the final step the background field comes from the result of the second step. All point data is now used, including grid-points illuminated by only a single radar. The tangential component for each radar observation is estimated by using the background from the previous step (i.e. non-radar data and/or multi-radar data).

The omega field is calculated by kinematically integrating the horizontal wind divergence. The lower boundary condition is specified by the surface wind and terrain gradient.

```
Auxiliary functions:
        write out graphical products


Inputs: ‘*’ = essential input
      * lga/fua grid           model data analysis/forecast
                               needed for current and previous cycle times
        pro     inter data     profiler, VAD, SODAR winds (all true north)
        snd     inter data     RAOB/Dropsonde data including winds
        pin     inter data     ACARS Winds (using pressure altitude)
        cdw     inter data     cloud drift wind
      * lso     inter data     LAPS surface obs file (e.g. mesonet & METAR)

        (remapped from raw radar data)
        v01-vxx grid           3-D radar reflectivity/radial velocity


Outputs: (LW3 is main output)
        pig        inter data   acars, cloud drift winds (prior to QC,
                                                   m/s, true north)
        prg        inter data   profiler, sounding winds (prior to QC,
                                                   m/s, true north,
                                               corrected for time)
        sag        inter data   surface winds  (prior to QC, true north)
        d01-dxx    inter data   derived radar vector obs    (grid north)

        lw3        3d grids     3-D winds (U and V are wrt GRID NORTH), omega
        lwm        2d grid      surface winds
```

Source directory: laps/src/wind (contains a README file)

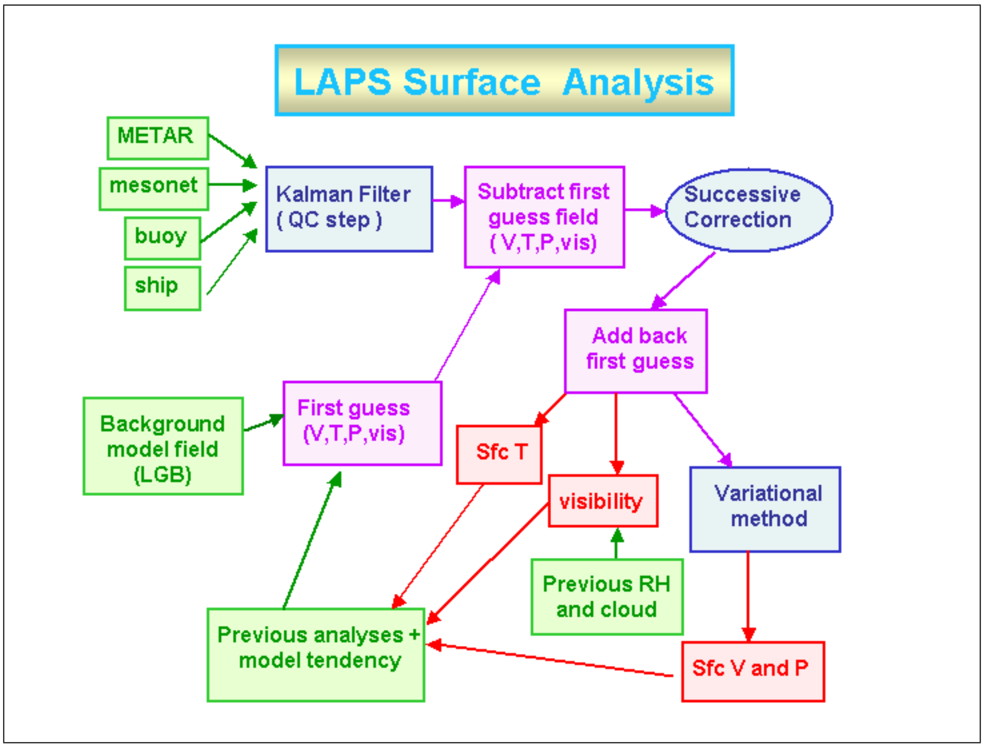

Figure 4.2: Flow chart of surface analysis processes

Parameter namelist files: 'static/wind.nl', 'static/nest7grid.parms' Further description and reference is at:

http://laps.noaa.gov/albers/laps/talks/wind/sld001.htm

4.3.2 SFC (LSX)

Surface processing - laps_sfc.x (LSX) (authors: John McGinley / Pete Stamus / Steve Albers)

The surface package collects surface data from the LSO intermediate data file (METARs, local mesonets via LDAD, buoy/ship obs), IR brightness temperatures, and fields from selected background models. Places surface data on LAPS grid and performs a simple quality control of the obs (climo + standard deviation checks). The quality control is described below in section 4.3.2.2. A flow chart can be seen at this URL:

http://laps.noaa.gov/albers/laps/talks/sfc/Sfc_anal.gif

and it is shown in figure 4.2. The background fields come from the locally-run LAPS model (FSF file), other large-scale models (RUC, ETA, AVN - via the LGB file), or a previous analysis (if all else fails). If the background model terrain is on a coarser grid than LAPS, this is accounted for so that the LGB fields have the fine-scale terrain related structure. For wind fields, the background comes from the 3-D wind interpolated to the surface or LWM file.

Prior to analysis of each field, another quality control step is done that rejects observations that deviate from the background by more than a threshold. This threshold is proportional to the standard deviation of the observation increments. The proportionality constant is set depending on the field.

The next step in the analyses is done with a successive correction technique similar to the 3-D wind and temperature analyses (see those sections and their web references). Observation increments are used for T, Td, U, V, MSL, P and straight observations are used for visibility. The temperature and dewpoint observations are also corrected for deviations of the station elevation from the LAPS terrain. Standard lapse rates are applied to this elevation difference. The analysis innovation is constrained to vary from the background by no more than the magnitude of the observation rejection threshold discussed above. This helps prevent overshooting (ballooning) of gradients into data sparse areas.

For relative humidity, the RH observations are converted into dew point using the station temperature (if the dew point isn't directly reported). The analyzed variable for moisture is dew point. After the analysis is performed the gridded dew point field is converted back into relative humidity using the analyzed temperature.

A land fraction term is factored into the weighting whenever the observation and grid point are on either sides of a 0.01 land fraction threshold. This helps prevent situations such as heating over the land having undue effects over the water areas. This weight is applied mainly to the T, Td, U, and V fields.

For pressure analysis, three fields are computed including reduced pressure (P) at reference height 'redp_lvl', surface pressure (PS), and mean sea level pressure (MSL). Background pressure fields come from the LGB or FSF files. The MSL background is used as read in upon input. The (PS) background is converted from the background model terrain to the LAPS terrain within the LGB/FSF file. The (P) background is generated by reducing the (PS) background to the reference analysis height 'redp_lvl' using Poisson's equation. This reference height should be approximately equal to the mean elevation of stations reporting surface pressure or station pressure.

Continuing the pressure analysis the altimeter setting observations are converted and added to the set of station pressures using the standard atmosphere. Station pressure observations are in turn converted to reduced pressure using Poisson's equation. The (P) analysis uses the (P) background plus the reduced pressure observation increments. The (P) analysis then uses variational techniques to constrain the surface winds and reduced pressures (P) to the full equations on motion. In contrast, mean sea level pressure (MSL) is a direct analysis of the MSLP observation increments together with the model background 'MSL' field. The station pressure analysis (PS) is calculated using the model background gridded 'PS' field, multiplied by the ratio of the (P) analysis to the (P) background.

Visibility is arrived at by first analyzing the surface visibility observations. A second step is applied to decrease the visibility in areas that have high RH and are near the cloud base that is given by the cloud analysis (in the previous time cycle).

Several derived variables are calculated before the LSX file is written. Also, a dependent data validation is done by interpolating several variables back to the observation locations and comparing the analysis to the obs. Output from this check is written to files located in

'$LAPS_DATA_ROOT/log/qc/laps_sfc.ver.hhmm', where 'hhmm' is the analysis 'systime'.

```
Inputs:
         LSO    surface observations
                    - or -
         LSO_QC QC'd surface observations

         LGA    Background model (3-D fields) on LAPS grid
                (State variables from 700mb and 500mb)

 LGB Background model (surface fields) on LAPS grid
                (TSF, PSF, SLP, DSF, P, VIS) fields
                    - or -
         FSF    Background local model (surface fields)
                (T, PS, MSL, TD, P, VIS) fields
                    - or -
         LWM    (background sfc wind from 3-D analysis - used for wind only)

         SND    Soundings (possibly containing surface obs)
         LC3    Cloud cover (for visibility)
 LM1    Soil moisture (for fire wx calc)
         LM2    Snow cover (for fire wx calc)

Output:
         LSX    LAPS surface data grids (24 2-d fields packed in one file)

                Includes various fields such as

         T, Td, Wind, MSLP, Reduced P (reference height sfc), Surface P

         Fire Danger:
             LAPS fire weather index is driven mainly by the surface fields
             of current humidity, wind, and temperature. RH and wind have
             the most weight with temperature having a lesser weight for
             this index that ranges from 0 to 20. Snow cover, elevation, and
             land fraction are given secondary consideration. High
             elevations, assumed to be above the treeline, are given a lower
             maximum value of 10. This index was developed primarily by
             Matt Kelsch of GSD.

         Colorado Severe Storms Index (CSSI):
```

```
    Severe storm potential mostly geared to the Colorado area. This
    uses a decision tree and various empirical functions. For more
    info please check the documentation in subroutine ‘make_cssi’
    in file ‘src/sfc/lapsvanl_sub.f’.

    Reference: Rodgers, D.M., and R.A. Maddox, 1981: Surface
    Meteorological Features Associated with Eastern Colorado Severe
    Convective Storms. NOAA Technical Memorandum ERL OWRM-13. DOC,
    NOAA, NWS, Boulder, Colorado, 21p.

    Link: \href{http://www.crh.noaa.gov/crh/?n=tm-114}

Heat Index:
    A function of temperature and humidity for discomfort due to heat.

    This is based on a formula from Lans Rothfusz, NOAA/NWS. It is
    calculated only when the surface air temperature exceeds 75 deg F.

    The idea is to give a "feels like" temperature.  For example, if
    the temperature is 85 F but the heat index is 100 F, most people
    would respond physically like it was 100 F actual temperature.
    It’s generally used to warn people that the temperature and
    humidity will combine to make it seem hotter than it actually is,
    and that they should take precautions like drinking more water,
    stay out of the direct sun, take frequent breaks if working
    outside, etc.

Ground Temperature:
    A single level analysis using land/sea surface (skin) temperature
    observations combined with background model information (if
    available).
```

Source directory: laps/src/sfc
Parameter namelist file: ‘static/surface_analysis.nl’

4.3.2.1 Surface Analysis Runtime Parameters

PRESSURE REDUCTION
You will need to select an elevation for the reduced pressure analysis. The reduced pressure is the only one really used in the variational portion of LAPS, and the idea is to select an elevation that is representative of the domain (or portion of the domain) you are interested in. For example, the Colorado LAPS domain includes 4000m high mountains over the western 1/3,

and plains that slope below 1000m at the eastern boundary. We use 1500m as the Colorado LAPS reduced pressure. This is close to the elevations over the eastern 2/3's of the domain, and requires a smaller reduction over the mountains compared to MSL, for example. Change the namelist variable in '/data/static/nest7grid.parms' when you localize LAPS.

4.3.2.2 Surface Analysis Quality Control

LAPS has a layered QC approach that gives us several opportunities to flag erroneous observations. To start with, a variety of gross "climo" checks are applied to the observations in the 'obs_driver.x' ingest program.

The next steps in quality control are encountered in 'laps_sfc.x'. This first checks the observations against climatologically reasonable ranges. Next, the observations (most fields except wind) are checked to see which ones are outliers (at 5 sigma) relative to the average observation value in the domain. As a further check, the Temperatures, Dewpoints and MSL pressures are checked to see if they deviate from the background field by more than a threshold absolute amount. The output from these checks is in both 'laps_sfc.log' and 'sfcqc.log'. The 'sfcqc.log' file contains the 'rely' (positive=retain, negative=reject) values designated as follows:

```
      |  STANDARD DEVIATION CHECK (against other obs)
      |
CLIMO |  PASS   N/A    FAIL
      |____________________________
      |
 PASS |   +35    10    -15
      |
 FAIL |   -99   -99    -99

  -25        failed model background comparison
  -99        observation was missing
```

If you wish to skip over these steps, you can change the 'surface_analysis.nl' namelist file. This is recommended when using the separate flag to use the experimental Kalman quality control observation file (lso_qc), generated by 'sfc_qc.exe'.

A new check looks at the temporal history of the obs where a 24 hour bias check flags temperature observations. Winds that aren't changing in speed or direction over the 24 period are also flagged.

There is an additional check for all analyzed fields (except visibility) within the 'spline' routine that rejects stations deviating from the background by more than a threshold number of standard deviations of the observation increments. This threshold can be independently adjusted (i.e. tightened or loosened) for each field via the 'surface_analysis.nl' namelist. If you see any bulls-eyes in the surface analysis that you don't believe, try contacting Steve Albers at GSD for more information on making these quality control namelist adjustments.

The experimental Kalman QC package 'sfc_qc.exe' (outlined in section 4.2.2.2) compares the observations temporally and fills in predicted values for an observation when it is only intermittently available. This helps compensate for temporal changes in data density. The log file for this new program is in 'sfc_qc.log'.

4.3.2.3 Surface Analysis Verification

Verification statistics for the surface analyses are written to the 'log/qc/laps_sfc.ver.hhmm' files. These contain information of obs differences relative to the background and the analysis. The obs listed have had most of the QC checks already applied, though an ob may have been rejected in the analysis by the final "spline standard deviation" check yet still have a non-missing value listed in the QC files.

4.3.2.4 STMAS-2D

Source directory: laps/src/mesowave/stmas_mg
Author: Yuanfu Xie (Yuanfu.Xie@noaa.gov)
Parameter namelist files: static/stmas3d.nl, static/nest7grid.parms

4.3.3 TEMP

Process: temp.exe - Temperature-Height analysis
Generate a temperature analysis using model background, sfc temp analysis, and RASS data. Quality control is applied to the temperature soundings. If any level in a sounding differs from the model background by more than a threshold (10 deg), the entire sounding is rejected.

```
Inputs: (from LGA, LSX, FUA [if available], LRS)       '*' = essential input
      * lga/fua grid            model data analysis / forecast
lrs inter data      RASS vertical temp profile
        snd     inter data      sounding temperatures (RAOB/Dropsonde/
                                                       Satellite Sounding)
pin inter data ACARS Temperatures (using pressure altitude)
      * lsx     grid            LAPS surface data grids

Outputs:
lt1 3d grid 3-D temperature (K), 3-D Heights (M-MSL)
        pbl     2d grid         2-D Boundary Layer Depth (m), and BL top (Pa)
        tmg     inter data      temperature obs used for lapsplot plotting
```

Source directory: laps/src/temp
Further description and reference is at:
http://laps.noaa.gov/albers/laps/talks/temp/sld001.htm

4.3.4 CLOUD

Process: cloud.exe - Cloud analysis package

Author: Steve Albers (Steve.Albers@noaa.gov)

Several input analyses are combined with METARs of cloud layers. These input analyses are the 3D temperature analysis, a three-dimensional LAPS radar reflectivity analysis derived from full volumetric radar data, and a cloud top analysis derived from GOES IR band eight data.

Vertical cloud soundings from METARs and pilot reports are analyzed horizontally to generate a preliminary three-dimensional analysis. This step provides information on the vertical location and approximate horizontal distribution of cloud layers.

The satellite cloud-top temperature field is converted to a cloud-top height field using the three-dimensional temperature analysis. The cloud-top height field is then inserted into the preliminary cloud analysis to better define the cloud-top heights as well as to increase the horizontal spatial information content of the cloud analysis. A set of rules is employed to resolve conflicts between METAR and satellite data. Finally, the three-dimensional radar reflectivity field is inserted to provide additional detail in the analysis.

```
Inputs:                                      '*' = essential input
      * lsx       grid    LAPS surface data grids
      * lt1           grid          LAPS 3-d temperature/height grid

      * vrc/v01/vrz  grid          2-D or 3-D radar reflectivity

      * lvd           grid          Infra-red and Visible satellite data
                                             (requirements are adjustable)
pin       inter data    pireps/clouds
        lm2           grid          composite snow cover (prev hour normally)
      * lga/fua       grid          model data analysis / forecast
      * lso           inter data    surface (e.g. METAR) obs

Outputs:
lc3     3d grid (ht)    3d clouds (fractional cover)
lps     3d grid         3D Radar Reflectivity (filled in)
lcb     2d grid cloud base/top (LCB,LCT) - all clouds are
                                considered (> .1 cover). Heights are MSL.
                                cloud ceiling (CCE) -  only areas analyzed
                                with a cloud fraction > 0.65 are considered.
                                Units are meters AGL.

lcv     2d grid column max cloud cover / snow cover
                                satellite/radar diagnostic fields
                                downward short wave radiation
```

Source directory: laps/src/cloud
Parameter namelist files: static/cloud.nl, static/nest7grid.parms
Further description and reference is at:
http://laps.noaa.gov/albers/laps/talks/cloud/sld001.htm

4.3.5 WATER VAPOR (Humidity Processing)

Last updated: 2/24/2006 by Daniel Birkenheuer
Code organization:
The moisture code is coordinated by the LQ3 modules all of which (with the exception of libraries) exist under ./src/humid/. The main driver, lq3driver.x contains only one subroutine call at this time.
./src/humid/lq3_driver1a.f (Module)
is the primary moisture processing module that sequences the various subroutines.
There is a second routine that formerly was used for HSM satellite processing; it is currently deactivated:
./src/humid/lq3_driver1b.f (Module) .
Now, using the CRTM forward radiance model and more advanced techniques, the satellite inclusion takes place in the above "1a" module. Treat the "1b" module as orphan code. Furthermore, a FORTRAN 90 compiler is required to fully compile the forward model along with the rest of the moisture analysis system.
Control file:
./data/static/moisture_switch.nl
An ASCII file intended for easy editing and control of the moisture modules activities. The first record controls usage of RAOB data (0=off, 1=on). The second record controls usage of satellite data (LVD files) and again (0=off; 8=on, use GOES-8, 9=on, use GOES-9). This module is exported with the RAOB feature OFF and the satellite feature ON and SET FOR GOES-8. The third switch enables (1) or disables (0) saturating air in cloudy areas. The fourth now enables using sounder data in lieu of imager data (GOES only). This should be set to (0) for the current time.
Data Particulars:
CLOUD DATA
A "switch" has been added to enable cloud data use to saturate air in cloudy areas. This is included as the last item in the moisture_switch.nl file that is maintained under the static area. To enable cloud data for saturating the air this is (1) to disable the feature, set the character to (0).
You might wonder why we need such a switch. During October (1996), we experienced problems with the cloud analysis. This was inadvertently causing problems in the moisture analysis through the cloud saturation adjustment. The incorrect moisture was in turn causing the models runs to fail. Hence we added this switch so that we could easily reactivate the feature once the cloud analysis was repaired without having to worry about recompiling any code.

RAOB DATA

The capability to ingest RAOB data into the moisture module has been available since 1996. There are two important items to know about:

1. The RAOB data are contained in lapsprd/snd/*.snd files. The moisture module will automatically use .snd data if present. If you do not wish to use sounding data there are 2 ways to exclude these data, the most obvious is to not provide .snd files.
2. In the event that you wish to exclude the use of sounding data and want them to be present in the data directory (possibly for some other application) you can avoid using them in the moisture code by modifying the file: ./data/static/moisture_switch.nl

The first record of this ASCII file is used for the RAOB data inclusion. The file itself is documented internally following the second record. If the first record is "1" (nominal case), the use of sounding data will be on, and .snd files will be processed if present. If this character is "0", the moisture code will not process sounding data.

Input files:

Inputs (status as of August 1996) ("grid" designates LAPS netCDF grid file unless otherwise stated):

```
*LGA/FUA     grid    MAPS/RUC background analysis or forecast (FUA)
 LSX         grid    LAPS surface analysis
 LC3         grid    LAPS 3-D clouds
*LT1         grid    LAPS 3-D temperatures
 SND         ASCII   RAOB observation file
 LVD         grid    Satellite data from AWIPS NOAAPORT/SBN
 LH1   grid    LAPS grid of VAS total precipitable water.
 LH2   grid    3 LAPS grids of VAS/radiometer modified precipitable
                      water.

Outputs (note LH3 contains 2 fields):

 LQ3         grid    3D Specific Humidity (floating point number)
 LH3         grid1   3D (RH3 field) Relative Humidity units of percent
                        0-100 (floating point number) with respect to
                        liquid water if ambient temperature is warmer
                        than 0 C, with respect to ice if ambient
                        temperature is equal to or less than 0 C
 LH3         grid2   3D (RHL field) Relative Humidity units of percent
                        0-100 (floating point number) with respect to
                        liquid water at all temperatures
 LH4         grid    2D Total precipitable water (meters) (floating point
                                number)
```

PRIMARY ALGORITHM SUMMARIES

RAOB ENHANCEMENT:

The RAOB data are added to the analysis via a second pass Barnes analysis. Normally, a Barnes analysis consists of two parts. The first fills the entire domain with values weighted by the distance to the neighboring points. In the second pass, a difference field (derived from the difference of the first pass and the observations) is added to the result from the first pass with adjusted weights to better tune to the scale of interest.

In this application we skip the first pass using instead the "background" analysis in place of the result of the first pass Barnes' result. The difference field is then generated and applied using a set of weights appropriate for the LAPS domain resolution and density of observations.

SATELLITE ALGORITHM:

An essential ingredient of the variational method is a satellite forward radiance model. The forward model produces a simulated radiance based on temperature, moisture, and ozone profiles along with the temperature of the surface or cloud top, and the pressure of that radiating surface (i.e., surface pressure or cloud top pressure whichever applies). Also needed are the zenith angle, used to determine the air mass path and optical depth between the radiator and the satellite. The forward model used for this work was obtained from NESDIS. The forward model coefficients used for this study were vintage late 1995.

In order to apply the forward model appropriately, a determination of clear and cloudy fields-of-view (FOV) need to be determined. The LAPS cloud analysis is used to identify clear and cloudy LAPS grid points. The analysis as presented here is only working from FOVs classified as clear. Cloudy FOVs probably can be used, but this is an early attempt at this technique, so a conservative approach was chosen. Later research may focus on using a combination of both clear and cloudy FOVs in the algorithm.

The first step in the algorithm is to assure all the data needed for proper execution are present. These include channel radiances derived from AWIPS imagery, the LAPS cloud analysis output, the LAPS surface temperature output, and LAPS 3-D temperatures. The forward model also requires an ozone profile along with moisture and temperature profiles above 100 hPa. These are gotten from climatology since LAPS extends only to 100 hPa. The entire ozone profile is provided by the forward model since LAPS does not analyze this parameter.

Next, the forward model is run to verify "clear" LAPS gridpoints, where clear is defined as those points in which both the modeled and measured GOES image radiances in channel 4 (11 micron) agree to with 2K. This step uses the LAPS thermal and as yet unmodified moisture profiles. Disparity in the channel 4 brightness temperature comparison indicates that the LAPS thermal profile is too far off or perhaps it is really cloudy where the LAPS cloud analysis is indicating it is clear. (It doesn't have to be totally cloudy for a disparity to exist, it can be partially cloudy and this will still be detectable in this difference test.) This is a conservative test; it really goes beyond simple cloud detection though that is a likely cause of differences, the forward model check is very sensitive and in many ways eliminates any thermal profiles that subsequent variational technique will find difficult to deal with. We are basically saying that we will not worry about moisture adjustment unless the thermal profiles are reasonable.

At this point, all gridpoints offering promise of moisture adjustment have been identified. If the domain is totally cloudy, the GOES adjustment is discontinued and returns unmodified moisture values which are passed to the final QC step. Assuming some gridpoints have been classified as clear, the next step is a variational adjustment at those locations. The functional evaluated at each gridpoint has and is best described in the literature (see articles under http://laps.fsl.noaa.gov/cgi/birk.pubs.cgi.

Basically a funcational is minimized that differences the perturbed solution against observation. The best perturbation is accepted as the "answer". The first term in the functional maximizes agreement between the forward model and observed radiance at the expense of only modifying the water vapor profile. The second term adds stability and gives more weight to solutions in which the coefficients departure from unity (no change to the initial profile) is minimized. The stability term was discovered to be necessary since without it some very good radiance matches were solved but with unreasonable coefficients.

Note that differences in all three channels are minimized in this technique, not only the moisture channel. Thus, any improvement in the "dirty window," channel 5, will also contribute to the solution. A variational technique is used to minimize this function and typically requires three to 10 iterations to converge. A limit of 50 iterations was set as the maximum number to attempt. If limit was reached, that particular gridpoint was excluded and treated as cloudy. Once the coefficients are determined, Laplace's equation is solved for interior points for which coefficients have not been determined. Then the entire domain is averaged using a spatial invariant filter; simply averaging the values in a 3x3 gridpoint window, assigning that average to the window's central grid location.

When the coefficients have been determined, they are applied to the specific humidity field at each pressure level for which they are designated. The modified specific humidity field is then advanced to the final analysis step.

Reference: http://laps.fsl.noaa.gov/cgi/birk.pubs.cgi

GPS ASSIMILATION ALGORITHM:

One of many terms in the humidity variational minimization step, the GPS total water is used to constrain the integrated water computed every iteration. Like the other terms in the functional used in the variational minimization, this term will reach a relative minimum when the state variables and specifically Q, best match this and other observations in a simultaneous manner (simultaneous here is respect to heterogeneous observational fit and not the more traditional state variable multi-variate solution sense).

The GPS algorithm traditionally read internal GSD netCDF files for input. It now has the capability (12/2010) to read MADIS surface data files for GPS data. The MADIS data are typically built every 5 minutes, so to read GPS data from these files, one should look back to the prior hour's MADIS file for the most recent GPS data. This is due to the fact that typically the GPS data are not ready for use until about Î© hour after acquisition time. So for a typical "20-min after the hour" LAPS run, the current hour's MADIS file will not contain any GPS data.

The software is currently tuned to open the prior hour's MADIS file and seek the "latest" GPS data that can be found in that file. This step is required since the MADIS file will be adding GPS data to it as it arrives and by the end of a given hour, MADIS files will contain 2 different GPS ingests.

Therefore the user, should be aware that if a LAPS start time other than 20-min past the hour is chosen, the software may have to be changed to acquire the most recent data. Right now things should be pretty stable in this regard. If one starts LAPS at the top of the hour, say 16 UT, this module will read the 15UT MADIS file for GPS data. It will likely find the "latest" data in that file to have been written about 15:20 UT. This is what the traditional read of internal GPS files would have returned.

Furthermore, if the LAPS system starts at 20-min after the hour, the same 15UT MADIS file will be opened and the GPS data read will be from about 15:45UT. Again, the routine will find the same that the traditional GPS file read would acquire. On the other hand, if one were to run at 15:50UT, there is a chance to miss the latest GPS data. The code as now written will open the 14UT MADIS file for data, when in fact the 15UT MADIS file may at this time contain data from 15:20UT. On the other hand, if this mistake is made, the data obtained will likely be nearly within the nearest hour of analysis time (14:50UT) and depending on the cycle time, may or may not be a critical issue. The user will have to determine whether this can be tolerated.

Cautions for STMAS: When reading GPS data in a 4DVAR context, GPS data reading will have to spend more time concerned with the actual data time associated with the GPS data. In this regard, multiple MADIS files will likely need to be opened, their contents matched with their respective observation times, and then the data will need to be temporarily stored, sorted, and processed according to the needs of 4DVAR.

4.3.6 DERIV

Process: deriv.exe - Derived products

Author: Steve Albers (Steve.Albers@noaa.gov)

These derived products are cloud, wind, stability, and fireweather related.

```
Inputs:                          '*' = essential input
      * lc3     3d grid (ht)     3d clouds (fractional cover)
      * lt1 grid LAPS 3-d temperature grid
lps     3d grid         3D Radar Reflectivity (filled in)
        lsx     2d grid          sfc pressure, temperature
lcv     2d grid column max cloud cover / snow cover
        lh3     grid             LAPS 3-d Relative Humidity (normally previous
                                                                        file)
        lso     inter data       Sfc (METAR) obs - for precip type verification

lw3 3d grid 3-D winds (U and V are wrt GRID NORTH)
        vrc/v01/vrz              2-D or 3-D radar reflectivity
```

```
Outputs:
lcp     3d grid (pres) 3d clouds (fractional cover) (pressure grid)

cty     3d grid         3D cloud type (CTY)
                                    threshold for cloud cover in CTY is 0.65.

pty     3d grid         3D precip type (PTY)

lwc     3d grids        Cloud liquid water content (LWC)
                        Cloud ice content (ICE)
                        Hydrometeor Concentration (PCN)
                                    Rain+Snow+Precipitation Ice Concentration
                        Rain Concentration        (RAI)
                        Snow Concentration        (SNO)
                        Precipitating Ice Conc.   (PIC)

                                The last four are specific contents in
                                kilograms/meter**3. These can be converted to
                                mixing ratio if desired by dividing through
                                by the air density.

lil     2d grid Vertically integrated cloud liquid water content
                                (lwc). This is the total cloud liquid condensed
                                in the column.

lct     2d grid SFC precip type (SPT,PTT)
                                    Types are:
                                       0 - No Precip
                                       1 - Rain             "R"
                                       2 - Snow             "S"
                                       3 - Freezing Rain    "Z"
                                       4 - Ice Pellets      "I"
                                       5 - Hail             "A"
                                       6 - Drizzle          "L"
                                       7 - Freezing Drizzle "F"
                                    SPT uses simple 0 dbz reflectivity
                                    threshold to define areas of precip.
                                    PTT uses a 13 dbz threshold for non-snow
                                    precip (~.01"/hr), 0 dbz is still used
                                    for snow though a surface dewpoint
```

```
                          depression threshold is used to filter out
                          areas of snow virga not reaching the ground.
                          The latter may be more useful for display
                          purposes by end users. PTT also utilizes
                          METAR data (in concert with analyzed cloud
                          fraction) to delineate areas of drizzle,
                          freezing drizzle, rain, freezing rain,
                          and snow - in areas that radar does not
                          detect echoes.

                      SFC cloud type (SCT)
                          This is the type of the lowest cloud layer
                          in the CTY (3-D cloud type) file. The
                          cover threshold is 0.65. The presence of
                          a CB higher up has priority. There are
                          10 possible cloud types.
                          Types are:
                             0 - No Cloud
                             1 - Stratus           "St"
                             2 - Stratocumulus     "Sc"
                             3 - Cumulus           "Cu"
                             4 - Nimbostratus      "Ns"
                             5 - Altocumulus       "Ac"
                             6 - Altostratus       "As"
                             7 - Cirrostratus      "Cs"
                             8 - Cirrus            "Ci"
                             9 - Cirrocumulus      "Cc"
                            10 - Cumulonimbus      "Cb"

lmd     3d grid         mean cloud drop diameter
lmt 2d grid         max echo tops (LMT),
                        Low level reflectivity (LLR)
lco     3d grid         Cloud omega - computed where cloud cover > .65
lrp     3d grid         3D icing index (integers 0-6)

                        0 is no icing
                        1 is light continuous
                        2 is mod   continuous
                        3 is heavy continuous
                        4 is light intermittent
                        5 is mod   intermittent
```

```
                              6 is heavy intermittent

    lst     2d grids          CAPE, CIN, and LI are calculated by lifting
                              a surface parcel taken from the LAPS surface
                              T and Td fields, as well as LAPS surface
                              (terrain following) pressure. LAPS 3-D
                              temperatures are also used.

                              CAPE (Convective Available Potential Energy)
                                  This is a net positive energy, so any
                                  negative area is subtracted from the
                                  positive area.

                              CIN (Convective Inhibition)
                                  Negative area in the sounding.

                              LI (Lifted Index)
                                  Environmental minus parcel temperature at
                                  500 mb.

                              SI (Showalter Index)

                              TT (Total Totals Index)

                              K (K Index)

                              LCL (Lifted Condensation Level)

                              WB0 (Wet Bulb Zero)

lhe 2d grid Helicity (Storm Relative Environmental). This
                              is integrated from the sfc to 3km AGL. It
                              is numerically equal to -2. times the hodograph
                              area.

                              A calculated storm motion vector is used
                              according to the Bunkers method used by the
                              National Weather Service. First a layer from
                              the sfc to 6km AGL is used to calculate the
                              mean wind. A shear vector through the sfc-6km
                              layer is also calculated. The storm motion
```

```
                                        vector is assumed to equal the mean wind vector
                                        plus a vector with a magnitude of 7.5 m/s and
                                        a direction 90 degrees to the right of the
                                        shear vector.

                                        Mean Winds (sfc - 6km layer)

liw 2d grid log(LI*Omega) (partially derived from 3-D winds)

lmr 2d grid 2-D column max radar ref

        lfr     2d grids        Fire weather indices as follows:
                                HAH  (High Level Haines Index)
                                HAM  (Mid Level Haines Index)
                                FWI  (Fosberg Fireweather Index)
                                VNT  (Ventilation Index)
                                UPB  (PBL Mean U Wind Component [grid-east])
                                VPB  (PBL Mean V Wind Component [grid-north])
                                CWI  (Critical Fire Weather Index)
```

Source directory: laps/src/deriv Further description and references are at:

http://laps.noaa.gov/albers/laps/talks/cloud/sld005.htm

and

http://laps.noaa.gov/albers/laps/talks/wind/sld007.htm

4.3.7 ACCUM

Process: accum.exe - Snowfall/Liquid Equivalent Precipitation
Author: Steve Albers (Steve.Albers@noaa.gov)

LAPS incremental/storm total snowfall/liquid equivalent accumulation.

```
Inputs:
      * lsx       grid LAPS surface data grids
      * lt1       grid LAPS 3-d temperature grid
        lh3           grid       LAPS 3-d Relative Humidity (normally previous)
        lso           inter data Rain gauge measurements of 1-hr precip

      * vrc/v01/vrz  grid       2-D or 3-D radar reflectivity

Outputs:
L1S          2d grid Snowfall over LAPS cycle time  (S01 field)
```

```
Storm total snow accumulation  (STO field)
                                Time interval is listed in the comment field.


Rain/Liquid Equivalent Precip  (R01 field)
                                over LAPS cycle time.


Storm Total Rain/Liquid Precip (RTO field)
                                Time interval is listed in the comment field.
```

Source directory: laps/src/accum

Parameter namelist file: static/nest7grid.parms

The precipitation analysis uses radar estimated precip rates as the primary dataset. The radar reflectiivty can be obtained from any combination of NOWRAD 2-D (section 4.2.4) or low-level reflectivity from mosaiced 2-D or 3-D radar reflectivity data. The source can be narrowband or wideband radar (section 4.2.2). The mosaics can be performed with either 2-D or 3-D inputs (section 4.2.4).

We presently use a Marshall Palmer Z-R relationship to obtain liquid equivalent precipitation. Snow is also estimated using a snow/rain ratio derived as a function of column maximum temperature. More on the basic accumulation processing is in Albers et. al., 1996. In the present LAPS version, areas without radar coverage switch over to a gauge only analysis of 1-hr precipitation - using a background or model first guess field (if available) or zero field as a first guess.

Areas having both radar and rain gauges present can be bias adjusted. An algorithm is presently be tested that determines this bias as a function of reflectivity.

Reference: Albers S., J. Mcginley, D. Birkenheuer, and J. Smart 1996: The Local Analysis and Prediction System (LAPS): Analyses of clouds, precipitation, and temperature. Weather and Forecasting, 11, 273-287 available at http://laps.noaa.gov/frd-bin/LAPB.pubs_96.cgi

4.3.8 SOIL MOISTURE

Process: lsm5.exe - Soil Moisture

Author: John Smart (John.R.Smart@noaa.gov)

LAPS soil moisture and snow cover

```
Inputs:
      * lsx      grid LAPS surface data grids
        l1s      grid LAPS surface precipitation
        lcv          grid       LAPS satellite derived snow cover


Outputs:
LM1           3d grid    Soil moisture (3-layer)
LM2           2d grids   Snow Cover and additional Soil Moisture info
```

This program is in the early stages of development and provides a three layer analysis of soil conditions. The three layers are as follows:

```
layer 1 (0-6in [0-0.152m])
layer 2 (6-12in [.152-.305m])
layer 3 (12-36in [.305-0.914m])
```

A snow cover analysis is included. The fractional snow cover is a composite over time of information from the cloud analysis (visible and IR satellite), and snow accumulation (derived mainly from radar). More documentation can be found within the source code (e.g. soilmoisture5.f, calc_evap.f).

Note that a soil temperature analysis is not included at this time. The closest thing we have to this is a single layer "ground temperature" analysis in the LSX surface output file.
Source directory: laps/src/soil

4.3.9 BALANCE

Process: qbalpe.exe - "Quasi-geostrophic balance of height, wind and clouds.
authors: John McGinley/John Smart/John Snook/Ed Tollerud, contact: Edward.Tollerud@noaa.gov

LAPS quasi-geostrophic balance of height and wind with temp adjustment. Cloud fields are now balanced with the other fields.

```
Inputs:
     * lw3          3d grid      LAPS wind analysis (grid north)
     * lt1          3d grid      LAPS height analysis
     * lsx          2d grid      LAPS sfc station pressure (PS field)
     * lwc          3d grid      LAPS Cloud Liquid/Ice/Precip
       lh3          3d grid      LAPS humidity
       lco          3d grid      LAPS Cloud Omega
     * lga          3d grid      Model first guess grids (including omega)


Outputs:
       lt1          3d grid    in lapsprd/balance/lt1 (ht and t field)
       lw3          3d grid    in lapsprd/balance/lw3 (u3, v3 and om field)
                                                      (grid north)
       lh3          3d grid    in lapsprd/balance/lh3 (rh field)
       lq3          3d grid    in lapsprd/balance/lq3 (sh field)
       lsx          2d grid    in lapsprd/balance/lsx (surface fields)
```

Source directory: laps/src/balance
Parameter namelist file: 'static/balance.nl'

The balance package starts by inputting the results from a simple, offline cloud model which retrieves liquid and ice partitioning and an estimate of vertical motion from the observed clouds (lwc/lco). The variational scheme is designed to accept cloud vertical motion estimates and ice and water content as observations. The cloud observations are fully coupled to the three dimensional mass and momentum field using dynamical constraints which minimize the local tendency in the velocities and ensure continuity is satisfied everywhere.

The scheme performs the analysis on the difference from an input model background with the benefit that existing background model balances need not be recreated each model cycle and that background model error daily compiled is input explicitly on a gridpoint by gridpoint basis.

Reference: McGinley, J.A. and J.R. Smart, 2001: On providing a cloud-balanced initial condition for diabatic initialization. Preprints, 18th Conf. on Weather Analysis and Forecasting, Ft. Lauderdale, FL, Amer. Meteor. Soc.

4.3.10 STMAS3D

Process: STMAS3D.exe - Space-Time Mesoscale Analysis System in 3D
This analysis can be run with the other appropriate executables by using the -V STMAS3D option in 'sched.pl'

4.4 Model Initialization & Postprocessing

The main GSD contact for information on how we use the LAPS analyses to initialize the forecasts is as follows: Chris Anderson for the RAMS/SFM/MM5/WRF/Eta models. The forecast models themselves are obtained separately from the LAPS analysis tar file. There is some documentation for the model interfacing at this URL:

http://laps.noaa.gov/mm5doc/README_lapsmm5.htm

4.4.1 LAPSPREP

Process: lapsprep.exe - Post-processes LAPS analysis files into formats that can be used to initialize a local forecast model (e.g., MM5, RAMS, WRF)
author: Brent Shaw, contact: Paul Schultz (Paul.J.Schultz@noaa.gov)

This process reformats LAPS data into files suitable for initializing a mesoscale forecast model. The output format is controlled by the "output_format" entry in lapsprep.nl and can be set to one of the following:

- output_format = 'wps' This causes the program to output a file in the WPS format (as needed for WRF version 3). Note that WPS has a constraint that the vertical levels of the initial condition (LAPS) be matched with those from the lateral boundary condition. This matching can be done either when running LAPS or in the WPS processing steps by three methods.

    1. In an example with the GFS as a lateral boundary condition one can reduce the levels in the LAPS 'static/pressures.nl' namelist as in this example

2. In an example with the NAM as a lateral boundary condition and if LAPS has more analysis levels than the NAM one can use the WPS utility 'mod_levs.exe'. It uses the 'namelist.wps' to rip out the NAM levels not in the namelist. The NAM data has levels from 1000 to 100 at a 25 mb interval plus a surface level.
3. The third method is the most desirable option that we recommend. We first run 'metgrid.exe' for the boundary conditions (e.g. NAM), starting at the initial time and proceeding through the forecast times. We then run 'real.exe' for the boundary conditions over the entire period. This will produce 'wrfbdy_d01' and 'wrfinput_d01' files. We next run 'metgrid.exe' for LAPS, followed by running 'real.exe' for LAPS only at the initial time. In this way the 'wrfinput_d01' (initial time file) will be overwritten by the LAPS initial condition.

- output_format = 'cdf' Writes a NetCDF file of the output
- output_format = 'wrf' This causes the program to output a file in the WRF Standard Initialization "grib_prep" format. These files can be read by the WRF SI "hinterp" process.
- output_format = 'mm5' This causes the program to output a file in the MM5v3 pregrid (v4) format that can be read in by MM5 the "regridder" pre-processor. See the NCAR MM5 REGRID documentation for the format specification of this output file.
- output_format = 'rams' This causes the program to output a file in the RAMS 4.x "RALPH2" format. These files can be read in by the RAMS ISAN pressure stage process. Note that RALPH2 files are in ASCII, so these files are actually human-readable. See the RAMS RALPH2 format specification for documentation.

There are three other namelist entries in the lapsprep.nl file:

1. hotstart: Set to '.true.' if you wish to include the cloud species from the cloud analysis in the output files. This currently only applies when output_format is equal to 'mm5' or 'wrf'.
2. balance: Set to '.true.' if you wish to use the wind and temperature, height, and surface analysis files from the balance package. This will only work if LAPS is running the balance package.
3. adjust_rh: Set to '.true.' if you wish to use the adjusted RH analysis from the balance directory.

This program essentially replaces part of the "dprep.exe" functionality, in that it produces initial conditions files for your local forecast model. If running a forecast model in real time, then this program should be executed immediately following the LAPS analysis during the hours in which the model will be initialized. It can simply be run as the last entry in sched.pl, which means you will always have an initial condition file avaialble immediately following your LAPS analysis.

To actually initialize a forecast model, you will still need to run the appropriate program to build the lateral boundary condition files, as LAPSPREP does not provide this function.

```
Inputs:          '*' = essential input
      * lw3                 grid      LAPS 3-d wind analysis (grid relative)
                                               U, V, Omega are written out
      * lt1                 grid      LAPS 3-d temperature & height analyses
      * lh3                 grid      LAPS 3-d relative humidity analysis
        lq3                 grid      LAPS 3-d specific humidity analysis
        lwc                 grid      LAPS 3-d cloud/precip hydrometeor analyses
      * lsx                 grid      LAPS 2-d surface analyses
        lm2                 grid      LAPS 2-d snow cover analysis

Outputs:
      mm5_init:YYYY-MM-DD_HH    grid   MM5 init. file (pregrid v3 format)
      ram_init:YYYY-MM-DD_HHMM  grid   RAMS init. file (RALPH2 format)
      wrf_init:YYYY-MM-DD_HH    grid   WRF init. file (gribprep format)
```

Parameter namelist file: 'static/lapsprep.nl'
Source directory: 'laps/src/lapsprep'

4.4.2 LAPS2GRIB

Process: laps2grib.exe - Converts LAPS analysis output into a single GRIB2 file located in the 'lapsprd/gr2' subdirectory. The parameters to convert are entered into a configuration file; the choice of parameters and the scaling of the parameters is controllable. (author: Brent Shaw)
Parameter namelist file: 'static/laps2grib.nl' lrun_laps2grib = .false. (default) or .true. (to create grib2 lapsprd/gr2 files)
Data file: 'static/laps2grib.vtab'
Source directory: 'laps/src/laps2grib'
E.g. parameters in data file: 'static/laps2grib.vtab' 3d 0,'lt1','t3 ', 1000.,110000., 1., 0,0, 0, 0 3d 0,'lt1','ht ', 1000.,110000., 1., 0,0, 3, 5 2d 'l1s','r01',1000.,4, 1, 0, 0,255,255,255,0, 1, 8
There are two numbers in the laps2grib.vtab file that immediately follow the file name extension and variable name: the conversion factor and the scale factor. The conversion factor will be multiplied by the LAPS variable coming out of the file in order to make the units conform to WMO specs (e.g. like cloud cover, which WMO defines as a percent from 0-100 whereas LAPS uses a fraction of 0-1., so we set the conversion factor to 100). In the case of precipitation, the units need to be in mm for GRIB, so if LAPS has precip specified in meters, then your conversion factor needs to be 1000 so you can get the data into mm. The scale factor specifies how many digits of precision to preserve after the decimal. It can be negative (for example, -1 would have precision to the nearest 10, 0 would give you to the nearest, and 1 gives you 1/10th, and so forth). So, if you have something that is typically very small (say, mixing ratio which in kg/kg ranges from 0.0001 to about 0.01, you might use a scale factor of 4 to preserve 4 digits after the decimal. On the other hand, with cloud cover you may only need the nearest integer value from 1-100, so you could use 0.

See laps/src/laps2grib/laps2grib.doc for more detailed information.

4.4.3 WFOPREP

Process: wfoprep.exe - Processes AWIPS/WFO large-scale model forecast files into formats that can be used as lateral boundary conditions to initialize a local forecast model (e.g., MM5, RAMS [SMF], WRF). The input files come from the SBN and are in netCDF format. (author: Brent Shaw)

This an optional program that can be used in the AWIPS/WFO environment. In other environments you'll want to use a different program to generate lateral boundary conditions.
Parameter namelist file: 'static/wfoprep.nl'
Source directory: 'laps/src/wfoprep'

4.4.4 LFMPOST

Process: lfmpost.exe - Post-processes WRF/MM5 model forecast files into formats that can be used to feed back into LAPS analysis or plotting software.
authors: Linda Wharton, Brent Shaw, John Snook, Steve Albers, Isidora Jankov, contacts: Linda Wharton / Steve Albers
Parameter namelist file: 'static/lfmpost.nl'
Source directory: 'laps/src/newlfmp' (new default version) 'laps/src/lfmpost' (old version)
The default (newer) version of the lfmpost program consists of a Fortran executable: $LAPSINSTALLROOT/bin/lfmpost.exe
This has been tested so far with WRF version 3.
There is also an "old" version of lfmpost. To build this version run 'make' and 'make install' in the 'src/lfmpost' directory. Then in $LAPS_DATA_ROOT/static (or the template) copy 'lfmpost_old.nl' to 'lfmpost.nl'.
LFMPOST is used to post-process raw model output files from the following models:

1. MM5 (Version 3 binary output format)
2. WRF (NCAR EM core, Version 1.3 netCDF output format) [old lfmpost.exe]
3. WRF (NCAR EM core, Version 2 netCDF output format) [old lfmpost.exe]
4. WRF (Version 3) [new lfmpost.exe]

It performs the following functions:

1. Read in model output for each time
2. Destagger variables to LAPS grid points
3. Vertically interpolation to isobaric levels
4. Output various formats, including LAPS fua/fsf netcdf format, Vis5D format, GRIB-1, and tabular text point forecast files.

It is controlled by the namelist file “lfmpost.nl”. If processing point forecasts, you also need to set up “lfmpost_points.txt”. Samples of these two files can be found in your LAPS_SRC_ROOT/data/static directory. To use lfmpost, you will need to copy these two files into MM5_DATA_ROOT/static or MOAD_DATAROOT/static (for MM5 or WRF, respectively) and edit them to your liking. If you are going to output LAPS fua/fsf files with lfmpost, you will need a valid LAPS_DATA_ROOT for the same model domain, and your pressure levels selected in lfmpost.nl must be the same levels selected in LAPS_DATA_ROOT/static/pressures.nl. Note that for this option, lfmpost expects the horizontal domain (dimensions, projection, etc.) to identically match for LAPS and the model being used.
To execute lfmpost, you should set the following environment variables as necessary:

- MM5_DATA_ROOT (if running MM5) MOAD_DATAROOT (if running WRF) LAPS_DATA_ROOT (if fua/fsf output is desired)

- LFMPOST expects to find the raw output files in: MM5_DATA_ROOT/mm5prd/raw (MM5) MOAD_DATAROOT/wrfprd (WRF v1 and v2)

Output from lfmpost goes into MM5_DATAROOT/mm5prd/d##/ for MM5 and goes into MOAD_DATAROOT/wrfprd/d##/ for WRF
Within the output directories, the following subdirectories need to exist to contain the specific output formats:
fsf -> For LAPS fsf files (2d and surface fields) fua -> For LAPS fua files (3d isobaric output) grib -> GRIB data [old lfmpost.exe] points -> Tabular text point forecasts v5d -> Vis5D files
After setting the appropriate environment variables and ensuring your namelists are configured properly, the syntax (old lfmpost.exe) is:
lfmpost.exe NAME DOMNUM where NAME is one of “mm5”, “wrf”, or “wrf2” for MM5, WRFv1.3, or WRFv2, respectively. DOMNUM is the nest to process.
Additional arguments are needed for the new (default) version of ‘lfmpost.exe’.
lfmpost.exe NAME WRFOUT NEST RCTIME FCSTTIME LAPS_DATA_ROOT NAME - one of “mm5”, “wrf”, or “wrf2” for MM5, WRFv1.3, or WRFv2, respectively.
WRFOUT - full filename of WRFout file
NEST - nest number (1 is outer)
RCTIME - CTIME in seconds of model initialization
FCSTTIME - number of seconds into the forecast
LAPS_DATA_ROOT - LAPS_DATA_ROOT where static files are set up (or the equivalent in the WRF directory)
LFMPOST is designed to operate on “incremental” raw model output data, so when you run WRF, be sure to output each time period to a separate file.
When running lfmpost in real-time for WRF output there is a Perl script that can be used: $LAPSINSTALLROOT/etc/lfmpost.pl (use wrfpost.pl for older lfmpost.exe)
There is also a driver script located in ‘etc/models/lfmpost_test.csh’, often used for non-realtime case runs, that can be executed as in this example:

./lfmpost_test.csh $LAPSINSTALLROOT $LAPS_DATA_ROOT $RUNTIME mvoutput
RUNTIME is model initialization time with format 'yyyymmddhh'

4.4.5 Forecast Graphics

A script can be run in cron (after the FUA/FSF files are created) to make GIF images of various forecast fields. This is located in 'etc/followup_fcst.pl'. Output images will appear in 'lapsprd/www/fcst2d'.

4.4.6 Verification

LAPS has a built-in verification package. This can be run after a model is run and the verifying observations and analyses are available. The driver script is in 'etc/verif/verif_fcst_driver.csh'. For real time runs it can be run via cron once for each model cycle. The script has several command line arguments that are described in comments at the top.
The script will produce stats files and PNG/GIF image output in the 'lapsprd/verif' directory tree. To help in setting this up or troubleshooting the results please note the input data that are being used:

1. FUA/FSF forecast files should be located in $LAPS_DATA_ROOT/lapsprd/f??/[model]/*.f??'
2. Observation and analysis files should be located in various other 'lapsprd' subdirectories. Some examples are as follows:

```
LSO - surface obs
LT1 - 3-D Temperature
LMR - Composite Reflectivity
LPS - 3-D Reflectivity
LCV - Solar Radiation (GHI)
```

3. Several parameters are relevant in 'static/nest7grid.parms' including:
   model_cycle_time, model_fcst_intvl, model_fcst_len, fdda_model_src
4. Log files are in $LAPS_DATA_ROOT/log/[*fcst*][*verif*]
5. Animated montages are an option that will work if 'followup_fcst.pl' is run prior to the verification (see previous sub-section 4.4.5)

APPENDIX A: Porting code mods from LAPS users back to GSD

We would like to encourage suggestions from LAPS users on how to improve LAPS, both scientifically and in the software itself. The changes should be made by downloading the most recent source code tree. Edit your changes in the source files, and then retar part or all of the source tree to send back to us. Please state the LAPS version number you had used. Any documentation pertaining to the reasoning behind the changes would be appreciated.

In some cases, a less formal process may be easier to go by. Here, the user can provide documentation of suggested mods either in descriptive form, or in terms of before and after code. The code author can then implement the changes in the GSD version. This can be useful in the event the mods are simple, or if the user has been working with a relatively old version of the software and/or there have been significant recent GSD mods to the software. This can also be useful if the user has an idea of a desired functionality within LAPS, but has not actually looked at the software details associated with implementing the functionality.

APPENDIX B: LAPS Output Variables and netCDF File Organization

LAPS Variables and netCDF File Organization (author: Linda.S.Wharton@noaa.gov).

LAPS output is written in netCDF format as summarized below. Each file extension goes into a separate directory under '$LAPS_DATA_ROOT/lapsprd/'. Note that netCDF information on the units of the fields, etc. is contained in the '$LAPS_DATA_ROOT/cdl/*.cdl' files.

```
                File LAPS CDF    Num
                Ext  Var  Var    Lvl  Field

Process surface:LSX  U    su        1  Surface (10m) wind u (grid north)
                     V    sv        1  Surface (10m) wind v (grid east)
                     P    fp        1  Reduced Pressure (constant height sfc)
                     PP   pp        1  Perturbation Pressure (if available)
                     T    st        1  Temp (2m)
                     TD   std       1  Dewpt Temp
                     TGD  tgd       1  Ground Temp (land surface/SST)
                     VV   vv        1  Vertical Velocity
                     RH   srh       1  Relative Humidity
                     MSL  mp        1  MSL Pressure
                     TAD  ta        1  Temp Advection
                     TH   pot       1  Potential Temp
                     THE  ept       1  Equivalent Potential Temp
                     PS   sp        1  Station Pressure (terrain following)
                     VOR  vo        1  Vorticity
                     MR   mr        1  Mixing Ratio
                     MRC  mc        1  Moisture Flux Convergence
                     DIV  d         1  Divergence
                     THA  pta       1  Potential Temp Advection
                     MRA  ma        1  Moisture Advection
```

```
                 SPD  spd       1  Surface Wind Speed
                 CSS  cssi      1  CSSI
                 VIS  vis       1  Surface Visibility
                 FWX  fwx       1  Fire Danger (LAPS / Kelsch)
                 HI   hi        1  Heat Index

Process temp:   LT1  T3   t        21  Temperature
                     HT   z        21  Height (geopotential meters)
                PBL  PTP  ptp       1  Boundary Layer Top (pressure)
                     PDM  pdm       1  Boundary Layer Depth (in meters)

Process accum:      L1S  S01  s1hr 1  Snow Accum Cycle
                         STO  stot 1  Snow Accum Storm Tot
                         R01  pc   1  Liq Accum Cycle
                         RTO  pt   1  Liq Accum Storm Tot

Process humid:  LQ3  SH   sh       21  Specific Humidity
                LH3  RH3  rh       21  Relative Humidity
                     RHL  rhl      21  Relative Humidity with resp to liquid
                LH4  TPW  tpw       1  Integrated Total Precipitable Water Vapor

Process wind:   LW3  U3   u        21  Wind u (wrt GRID NORTH)
                     V3   v        21  Wind v (wrt GRID EAST)
                     OM   om       21  Wind omega
                LWM  SU   u         1  Surface wind u (wrt GRID NORTH)
                     SV   v         1  Surface wind v (wrt GRID EAST)
Process cloud:   LC3  LC3  camt    42  Fractional Cloud Cover (levels 1-42)
                 LCB  LCB  cbas     1  Cloud base
                      LCT  ctop     1  Cloud Top
                      CCE  cce      1  Cloud Ceiling
                 LCV LCV  ccov      1  Cloud Cover
                     CSC  csc       1  Cloud Analysis Implied Snow Cover
                     ALB            1  LAPS derived albedo
                     S3A            1  3.9u satellite data
                     S8A            1  11u satellite data
                     RQC            1  Radar QC information (2D vs 3D)
                     SWI            1  Downward Shortwave Radiation
                LPS  REF  ref      21  LAPS Radar Reflectivity

Process deriv:  LCP  LCP  ccpc     21  Fractional Cloud Cover Pressure Coord
                LWC  LWC  lwc      21  Cloud Liquid Water
```

```
     ICE  ice     21  Cloud Ice
     PCN  pcn     21  Hydrometeor Concentration
     RAI  rai     21  Rain Concentration
     SNO  sno     21  Snow Concentration
     PIC  pic     21  Precipitating Ice Concentration
LIL  LIL  ilw      1  Integrated Liquid Water
LCT  PTY  spt      1  Sfc Precip Type
     PTT  ptt      1  LAPS Sfc Precip Type
     SCT  sct      1  Sfc Cloud Type
LMD  LMD  mcd     21  Mean Cloud Drop Diameter
LCO  COM  cw      21  Cloud omega
LRP  LRP  icg     21  Icing Index
CTY  CTY  ctyp    21  Cloud Type
PTY  PTY  ptyp    21  Precip Type

  LMT  LMT  etop   1  Max Echo Tops
       LLR  llr    1  Low Level Reflectivity
LST  LI   li       1  Lifted Index
     PBE  pbe      1  Positive Bouyant Energy
     NBE  nbe      1  Negative Bouyant Energy
     SI   si       1  Showalter Index
     TT   tt       1  Total Totals Index
     K    k        1  K Index
     LCL  lcl      1  Lifted Condensation Level
     WB0  wb0      1  Wet-Bulb Zero
LWM  SU   u        1  Surface wind u (grid north)
     SV   v        1  Surface wind v (grid east)
LHE  LHE  hel      1  Helicity
     MU   mu       1  Mean wind u (grid north)
     MV   mv       1  Mean wind v (grid east)
LIW  LIW  liw      1  log(LI*omega)
     UMF  umf      1  Upslope Component of Moisture Flux
LMR  R    mxrf     1  Column Max (Composite) Radar Reflectivity
LFR  HAH  hah      1  High Level Haines Index
     HAM  ham      1  Mid Level Haines Index
     FWI  fwi      1  Fosberg Fireweather Index
     VNT  vnt      1  Ventilation Index
     UPB  upb      1  PBL Mean Wind U-component (grid north)
     VPB  vpb      1  PBL Mean Wind V-component (grid east)
     CWI  cwi      1  Critical Fire Weather Index
```

```
Process soil:   LM1  LSM  lsm      3  Soil Moisture
                LM2  CIV  civ      1  Cumulative Infiltration Volume
                     DWF  dwf      1  Depth to wetting front
                     WX   wx       1  Wet/Dry grid point
                     EVP  evp      1  Evaporation Data
                     SC   sc       1  Snow covered
                     SM   sm       1  Snow melt
                     MWF  mwf      1  Soil Moisture content Wetting Front
LAPS Fcst Model:
                FUA  U3   ru      21  Fcst Model Wind u (grid north)
                     V3   rv      21  Fcst Model Wind v (grid east)
                     HT   z       21  Fcst Model Height (geopotential meters)
                     T3   rt      21  Fcst Model Temperature
                     SH   rsh     21  Fcst Model Specific Humidity
                FSF  USF  usf      1  Fcst Model Surface wind u (grid north)
                     VSF  vsf      1  Fcst Model Surface wind v (grid east)
                     TSF  tsf      1  Fcst Model Surface Temperature
                     DSF  dsf      1  Fcst Model Dewpoint
                     RH   rh       1  Fcst Model Relative humidity
                     LCB  lcb      1  Fcst Model Cloud base
                     LCT  lct      1  Fcst Model Cloud top
                     P    p        1  Fcst Model 1500m pressure
                     SLP  slp      1  Fcst Model MSL pressure
                     PSF  psf      1  Fcst Model Surface pressure
                     LIL  lil      1  Fcst Model Integrated cloud liquid water
                     TPW  tpw      1  Fcst Model Total precipitable water vapor
                     R01  r01      1  Fcst Model Liquid accum cycle
                     RTO  rto      1  Fcst Model Liquid accum storm total
                     S01  s01      1  Fcst Model Snow accum cycle
                     STO  sto      1  Fcst Model Snow accum storm total
                     TH   th       1  Fcst Model Potential temperature
                     THE  the      1  Fcst Model Equivalent potential temp
                     PBE  pbe      1  Fcst Model Positive buoyant energy
                     NBE  nbe      1  Fcst Model Negative buoyant energy
                     PS   ps       1  Fcst Model Surface pressure
                     CCE  cce      1  Fcst Model Cloud ceiling
                     VIS  vis      1  Fcst Model Visibility
                     LCV  lcv      1  Fcst Model Cloud cover
                     LMT  lmt      1  Fcst Model Max echo tops
                     SPT  spt      1  Fcst Model Sfc precip type
                     LHE  lhe      1  Fcst Model Helicity
```

```
                LI   li        1  Fcst Model Lifted index
                HI   hi        1  Fcst Model Heat index
                SWO  swo       1  Fcst Model Outgoing Shortwave Radiation
                LWO  lwo       1  Fcst Model Outgoing Longwave Radiation
                FWI  fwi       1  Fcst Model Fosberg fire weather index
                FWX  fwx       1  Fcst Model Kelsch fire weather index
           RSM  LSM  lsm      11  Fcst Model Soil Moisture

Intermediate LAPS files:
Process vrc_driver:
           VRC  REF  ref       1  NOWRAD 2D radar reflectivity

Process mosaic_radar:
           VRZ                21  (3D reflectivity mosaic)

Process remap:  V01  REF  refd    21  Radar reflectivity
                VEL  veld    21  Radial Velocity
                NYQ  nyqd    21  Nyquist velocity
           files V02, V03, V04, V05, V06, V07, V08, V09, V10, V11,
                 V12, V13, V14, V15, V16, V17, V18, V19, V20 same format

Process lga.exe (background model):
           LGA  HT   ht       21  Model isentrop height interp to LAPS
                                  isobaric (geopotential meters)
                T3   t        21  Model isentrop temp interp to LAPS isobaric
                SH   sh       21  Model specific humidity
                U3   u        21  Model u wind component (grid north)
                V3   v        21  Model v wind component (grid east)
                OM   om       21  Model vertical velocity (Pascals/second)
           LGB  USF  usf       1  Model Surface wind u (grid north)
                VSF  vsf       1  Model Surface wind v (grid east)
                TSF  tsf       1  Model Surface Temperature
                TGD  tgd       1  Model Ground Temperature
                DSF  dsf       1  Model Dewpoint
                SLP  slp       1  Model MSL pressure
                PSF  psf       1  Model Surface pressure
                RSF  rsf       1  Model Specific Humidity
                P    p         1  Model reduced pressure
                PCP  pcp       1  Model Precipitation

Process LH1:    LH1  PW   pw  1  Precipitable Water Vapor
```

```
Process LH2:         LH2  PW   lpw 3  Layer Precipitable Water Vapor

Process lvd_sat_ingest:
                     LVD  S8W  s8w 1  GOES IR band-8 bright temp warmest pixel
                     S8C  s8c      1  GOES IR band-8 bright temp coldest pixel
                     SVS  svs      1  GOES visible satellite - raw
                     SVN  svn      1  GOES visible satellite - normalized
                     ALB  alb      1  albedo
                     S3A  s3a      1  GOES IR band-3 bright temp averaged
                     S3C  s3c      1  GOES IR band-3 bright temp filtered
                     S4A  s4a      1  GOES IR band-4 bright temp averaged
                     S4C  s4c      1  GOES IR band-4 bright temp filtered
                     S5A  s5a      1  GOES IR band-5 bright temp averaged
                     S5C  s5c      1  GOES IR band-5 bright temp filtered
                     S8A  s8a      1  GOES IR band-8 bright temp averaged
                     SCA  sca      1  GOES IR band-12 bright temp averaged
                     SCC  scc      1  GOES IR band-12 bright temp averaged
                     Note: band-8 is approx 11.2 microns.

Static LAPS file - run by localization:
gridgen_model.exe: creates file ‘static.nest7grid’
                     LAT           1  Latitude (degrees)
                     LON           1  Longitude (degrees)
                     AVG           1  Mean elevation MSL (m)
                     STD           1  Unused
                     ENV           1  Unused
                     ZIN           1  Z coordinate - used for plotting in AVS
                     LDF           1  Land Fraction (0=water,1=land)
                     USE           1  Landuse
```